AN ARGUS SPECIALIST PUBLICATION

JULY 1985 £1.00

COMPUTING Today

CRIBBAGE
new series on programming

PASCAL
Oxford Pascal on test

MAPLE
mapping landscapes with micros

Z800
latest on Zilog's next processor

- MAXAM -
FOR THE AMSTRAD

THE COMPLETE CODE DEVELOPMENT SYSTEM FOR THE AMSTRAD CPC 464.
★ ASSEMBLER ★ MONITOR ★ TEXT EDITOR ★

"The Arnor system is the best editor/assembler to be released for the AMSTAD so far" - PCN 100
"For flexibility and ease of use, ARNOR is easily the best I have seen" Pop. C. Wkly Vol 4 No 8
"assemblers.... look no further, ARNOR's is the best I have seen... by far the easiest to use and most friendly I have come across"
- Computing with the Amstrad. April 1985.

Now available in **ALL THREE** formats

Tape (only) £13.50

Disc £26.90.

16K ROM + multifunction adaptor £59.90

(All prices include VAT, p & p)

Cheques/PO's to ARNOR Ltd Dept CD
PO BOX 619, London SE25 6JL
-Overseas - no extra - Trade enq's welcome -

Make MAXimum use of your AMstrad

Hotline 01.653.1483 **Technical 01.852.2174**

THE MALTRON KEYBOARD
Why Hasn't Anyone Thought Of It Before?

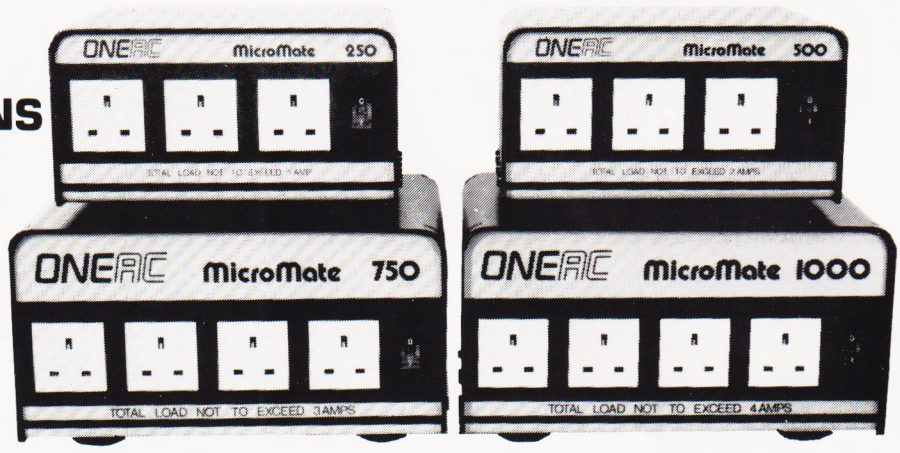
If you spend a lot of time at your keyboard it is very likely that you get unpleasant aches and pains that can extend from your wrists to your neck and shoulders.
The cause is the shape and letter layout of the standard QWERTY keyboard, whose basic design goes back to the last century.
The MALTRON keyboard is easy to learn and tests have shown that great improvements in efficiency and comfort can be obtained.

Contact: **P.C.D. MALTRON LTD**
15 Orchard Lane,
East Molesey,
Surrey KT8 0BN
Tel: 01-398-3265

The copyright subsisting in this keyboard is the property of P.C.D. Maltron Ltd.

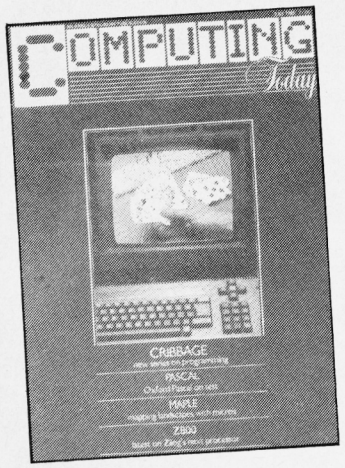
WHAT DOES "ONEAC" MEAN?
COMPUTER FRIENDLY!

OPTIMIZED MAINS CONDITIONING THROUGH YOUR ONEAC POWER PROFESSIONAL AT:



ONEAC® LIMITED
Unit 6, Eyston Way,
Abingdon, Oxon, OX14 1TR

Abingdon (0235) 34721
Telex: 837976 ONEAC G.



COVER

Techniques for games programming are explored this issue, and for the next four months with Our Cribbage Player series, beginning on page 26.

CONTENTS

VOL 7 NO 7 JULY 1985

REGULARS

NEWS 6

COURSES AND TRAINING 12

PRINTOUT 17
Readers' letters.

ALGORITHM ANGLES 31
Solving puzzles.

FIVE YEARS BACK 16

BOOK PAGE 46
Garry Marshall looks at four books on Expert Systems.

SERIES

LESSONS OF HISTORY 18
The transistor comes of age.

CRIBBAGE PLAYER 26
Discover the workings of a Cribbage game for the Amstrad CPC464/664 and the BBC micro.

LEARN 'C' 38
Mark Woodley concludes this series on C programming.

GENERAL FEATURES

MAPLE 22
Mapping landscapes with micros.

POWERING UP 36
The technology that keeps your micro running.

THE ZILOG Z800 42
The latest on the Z80's 16-bit brother.

INSIDE FASTLOADERS 50
How fastloaders work.

A LOOK INTO THE FUTURE 52
A considered opinion of the future for computing.

MULTIBUS 54
Bill Horne explores one of the most popular racking standards.

SPECIAL FEATURES

KUMA'S ARTWORK 21
Kuma Computers' graphics package for the Amstrad reviewed.

THE ARNOR MAXAM 32
We look at a very nice assembler for the Amstrad CPC464/664.

OXFORD PASCAL 40
Eriklas Pittas reviews Oxford Pascal for the BBC micro.

TRAPPING ZEROES 34
Control out-of-memory errors on the BBC micro with this ingenious utility.

CLUBFILE 56
A multipurpose database program for Atari micros.

Subscriptions 4
Backnumbers 5
Classified Advertisements 63
Next month's Computing Today 67

Editor: Don Thomasson
Assistant Editor: Jamie Clary
Technical Illustrator: Jerry Fowler
Additional Illustration: Grant Robertson
Advertisement Manager: Anthony Shelton
Classified Sales Executive: Caroline Falkner
Advertisement Copy Control: Sue Couchman, Lynn Collis
Publishing Director: Peter Welham
Chairman: Jim Connell
Origination and Design: Design International
Cover Design: Argus Design

Member of the Audit Bureau of Circulation
ISSN 0142-7210

Computing Today is normally published on the second Friday in the month preceeding the cover date. Distributed by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, Lodnon EC2A 4JS. 01-247 8233. Printed by: Alabaster Passmore & Sons Ltd, Maidstone, Kent.

The contents of this publication including all articles, designs, plans, drawings and programs and all copyright and other intellectual property rights therein belong to Argus Specialist Publications Limited. All rights conferred by the Laws of Copyright and other intellectual property rights and by virtue of international copyright conventions are specifically reserved to Argus Specialist Publications Limited and any reproduction requires the prior written consent of the Company. © 1985 Argus Specialist Publications Limited.

Subscription notes; UK (£16.20) including postage. Airmail and other rates upon application to Computing Today Subscriptions Department, Infonet Ltd., Times House, 179 The Marlowes, Hemel Hempstead, Herts. HP1 1BB England (phone 0442 48432).

EDITORIAL & ADVERTISEMENT OFFICE: 1 Golden Square, London W1R 3AB. Telephone: 01-437 0626. Telex: 8811896.

Subscriptions

Personally, we think you'll like our approach to microcomputing. Each month, we invite our readers to join us in an abundance of feature articles, projects, general topics, news and reviews – all to help committed micro users make more of their microcomputers at home or at work.

However, if you've ever missed a copy of Computing Today on the newstands, you'll not need us to tell you how valuable a subscription can be. Subscribe to CT and for a whole year you can sit back, assured that each issue, lovingly wrapped, will find its way through your letter box.

And it's not difficult! All you have to do is fill in the form below, cut it out and send it (or a photocopy) with your cheque or Postal Order (made payable to ASP Ltd) to:

COMPUTING TODAY Subscriptions,

Infonet Ltd,
Times House,
179 The Marlowes,
Hemel Hempstead,
Herts HP1 1BB.

Alternatively, you can pay by Access or Barclaycard in which case, simply fill in your card number, sign the form and send it off. Please don't send in your card.

Looking for a magazine with a professional approach with material written by micro users for micro users? Why not do yourself a favour and make 1985 the year you subscribe to Computing Today and we'll give you a truly personal approach to microcomputing.

SUBSCRIPTION ORDER FORM

Cut out and SEND TO :

COMPUTING TODAY Subscriptions
INFONET LTD,
TIMES HOUSE,
179 THE MARLOWES,
HEMEL HEMPSTEAD,
HERTS HP1 1BB.

Please commence my subscription to Computing Today with the ..... issue.

SUBSCRIPTION RATES (tick ☐ as appropriate)

£16.20 for 12 issues UK	☐
£18.70 for 12 issues Overseas Surface	☐
£51.20 for 12 issues Overseas Air Mail	☐

I am enclosing my (delete as necessary) cheque/ Postal Order/ International Money Order for £.....
(made payable to ASP Ltd)
or
Debit my Access/ Barclaycard*
(*delete as necessary)

Please use BLOCK CAPITALS and include postcodes.

NAME (Mr/ Mrs/ Miss) .....
delete accordingly

ADDRESS .....

.....

..... POSTCODE .....

Signature .....

Date .....

BACKNUMBERS

JANUARY 1984
TRS-80 programmer's aid, Apple music, Electron review, TRS-80 screen editor, calendar program.

FEBRUARY 1984
Using MX-80 graphics, Colour Genie monitor, non-random random numbers, ZX81 Forth, Program recovery on the Commodore 64.

MARCH 1984
Easycode part 1, BBC poker, Spectrum SCOPE review, Genie utilities, Spectrum Centronics interface.

APRIL 1984
MEMOTECH MTX500 review, Genie BASIC extensions, Brainstorm review, Disassembly techniques, Recursion.

MAY 1984
Debugging, Spectravideo SV318 review, Extending the Commodore 64's BASIC part 1, Z80 text compactor.

JUNE 1984
Adler Alphatronic review, Digithurst's Microsight review, Commodore search and replace, CP/M directory, Interrupts.

JULY 1984
Commodore BASIC extensions reviewed, The Art of Islam, a fast sort, Brother HR5 review, Random Thoughts, extended palette on the Dragon.

AUGUST 1984
Apricot xi review, BBC Mode 7 screen editor, Genie sprites, Microdrive-file line editor, TRS-80 screen scroller.

SEPTEMBER 1984
CUBE's Beebflex, Electron drawing utility, MTX real time clock, Commodore SX64 review, BBC disassembler, TRS-80 Fastsave.

OCTOBER 1984
AMSTRAD CPC464 review, Dragon sprites, Commodore 64 adventures, BBC Draughts, Nascom screen dump.

NOVEMBER 1984
Apple IIc review, Epson PX8 review, MTX utilities, Z80/TRS-80 memory move routine, 16-page Business supplement.

DECEMBER 1984
Acorn Bitstick package review, Art and the AMSTRAD, BBC Draw, Psion Organiser review, Koala Pad review.

JANUARY 1985
BBC Commodities, Tatung Einstein review, Fujitsu Micro 16 review, Commodore 64 prettyprint, MTX500 Life, Nascom string-save.

FEBRUARY 1985
The Intelligent Computer, Dragon interrupts, BBC Machine-code monitor, Tasword 464 review, Spectrum/BBC cassette volume meter, Sakata SCP800 printer/plotter review, Spectrum ON ERROR, TRS-80 mail list, BBC passwords; Deficiency, Abundance, Perfection.

If you've lost, lent or had stolen one of those precious back copies of Computing Today then now is your chance to fill the gap in your collection. The list of issues given here represents the few remaining copies that we have available to help complete your library of all that's good in features, programs and reviews.

If you want one of these issues, it's going to cost you **£1.40** (including postage and packing) but we think that's a small price to pay for the satisfaction you'll get. Ordering could hardly be made simpler — just fill in the form, cut it out (or send a photocopy) together with your money to:

Backnumbers,
Infonet Ltd,
Times House,
179 The Marlowes,
Hemel Hempstead,
Herts HP1 1BB.

If you wait until next month to do it, the chances are that we'll have run out of the very issue you wanted!

A COMPUTING Today Reader Service

BACKNUMBERS

Please send me the following Backnumbers

ISSUE

At £1.40 each. I enclose £.....

NAME .....

ADDRESS .....

.....

.....

..... POSTCODE .....

Signature .....

I enclose a cheque/PO for £.....(Payable to ASP Ltd)
I wish to pay by credit card

Access ☐ Barclaycard ☐

Buy it with Access

BARCLAYCARD VISA

Insert Card No.

If you wish to pay by Access or Barclaycard, just fill in your card number and sign the form. **do not send your card.**

Please allow 21 days for delivery.

NEWS

CHIP BAULKS HACKERS

Intel's new 128K (16K x 8) Keyed Access EPROM employs a data protection mechanism to increase software security and also prevent systems from executing unauthorised software.

Known as the 27916 KEPROM*, the device is pin-compatible with the 27128 EPROM and can be configured in three different ways to allow its special features to be customised for individual application requirements.

In the KEPROM configuration (selected by programming a special EPROM bit), the 27916 prevents improper access by implementing the protection mechanism that locks-out unauthorised users. Only those authorised to access the device's contents — by using an initialisation sequence called Authentication Handshake — can freely read the stored data.

The Handshake involves two KEPROM-configured devices that communicate with each other according to a verification protocol. Initially, one device sends its encrypted key to the second device for comparison; if the comparison is true, the KEPROMs change roles and the procedure is repeated. If the second comparison is unfavourable, the data in both devices is unreadable.

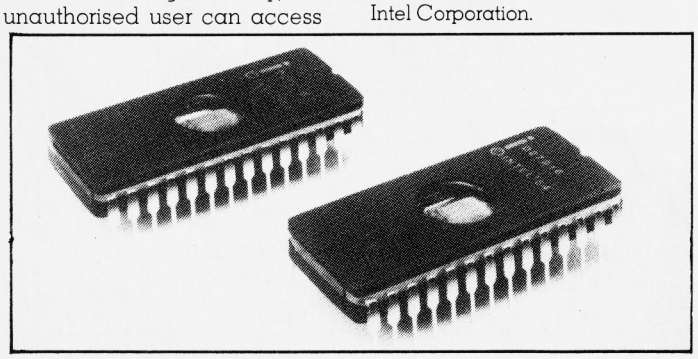
All KEPROMs contain encryption circuitry which implements a proprietary logic combination of a random number and a Key. The Key is a confidential user-unique 64-bit number, programmed into a special portion of the device; without knowing this Key, no unauthorised user can access the memory array. Since the Key provides ultimate security of the software, it cannot be read and is known only to the system/software designer.

The second 27916 configuration, 'Key Manager', is used in a system where there are many KEPROMs. This special configuration can accomplish the Authentication Handshake with differently-Keyed KEPROMs and unlock only those which are authorised to be used in the Key Manager's system. The Key Manager is capable of handshaking with up to 1024 different Keys.

In the third configuration, '27128', the 27916 device is fully erasable and re-programmable, and resembles its 27128 EPROM-only counterpart in every respect. All standard 27128 modes are available for use in this configuration.

The 27916 is fabricated using Intel's HMOS-E process and has a JEDEC-approved industry-standard pinout. Device characteristics include a 250ns access time; compatibility with the 8 MHz iAPX-186 processor and 12 MHz 8051 family; and an intelligent Programming Algorithm for fast EPROM programming. The 27916 also has a standby mode which reduces power consumption without increasing access time — standby current is only 40mA. □

* KEPROM is a trademark of the Intel Corporation.



COMAL FOR BBC MICRO

COMAL, the internationally approved educational microcomputer language, is now available for the BBC Micro in a ROM-based version from Acornsoft.

COMAL, a high-level language similar to BBC BASIC and Pascal (but drawing onthe better features of both), has already been chosen by a number of European countries, including Ireland, Denmark and Sweden, as their official language for secondary school computing. It is also approved for use in Scotland, and the USA is also proving an enthusiastic market.

Unique amongst computer languages, COMAL is being nurtured under the guidance of 'Standardisation and Development Groups' — twice-yearly meetings of all parties concerned with COMAL's application, implementation, improvment and conformity between versions.

COMAL is user-friendly and minimises bugs by drawing attention to syntax errors as lines are entered. It also checks structure before the program is run. A block structured language, COMAL is said to naturally promote good programming practice. It also provides extensive parameter passing facilities for procedures and functions, supports reals, strings and multi-dimensional arrays, and simplifies file I/O through excellent file-handling facilities.

Acornsoft COMAL comes on a 16K ROM for the BBC Micro, and includes a very comprehensive 440 page user guide with sections for both the beginner and the experienced programmer. Price: £49.95 incl VAT. Extra User Guides: £10.00 each. □

MICRONET CHARGES UP BUT NEW SERVICE OFFERED

Micronet 800 have raised subscription charges from £8 per quarter to £10 per quarter.

This is the first price rise since the launch of the service in March 1983, but the past two years have seen many enhancements to the service. The extension of local call access now enables subscribers to benefit from substantially lower running costs, and the mailbox service became nationally available last October, allowing customers to exchange electronic mail at local call rates.

Coming shortly will be the Gallery, allowing any Micronet member to have complete editorial control of a page of up to 26 frames on the Micronet database and in the Autumn Chatline services will become mainframe-based providing a real-time, subject-divided and faster Chatline service.

ZILOG PREPARE RELEASE OF SUPER 8 SINGLE CHIP MICRO

The Zilog Z8 family of single chip microcomputers is to get a new member — the Super 8.

The present Z8 range consists of several MCUs, among them the Z8601 with 2K ROM, and the Z8603 prototyping device with EPROM interface.

The Super 8 is an enhanced specification Z8, and Zilog's 'Rumour Control' department inform us that although the first silicon prototype hasn't worked — something not too unusual — further iterations are expected soon.

An interesting fact, is that an iteration scheduled for November of this year is to carry a Forth Kernal, and we are told that a Super 8 running Forth will significantly out-perform a Z80 running machine code.

Could Zilog be planning a new push on Forth? Zilog's John Chewter predicts a new 'Forth explosion' in the near future, although it is difficult to guage the extent of Zilog's activities.

Prelininary specifications for the Super 8 feature in the latest Zilog Component Data Book, but no release date is suggested.
JC

Roman Iwanczuk is to be Zilog's Super 8 specialist: read his article on the Z800 family this issue.

FIG FORTH FOR THE AMSTRAD

Kuma Computers have now released a version of FIG Forth for the Amstrad CPC464.

Kuma Forth for the CPC464 is a comprehensive implementation of FIG Forth with many features added to allow extensive use of the facilities provided by the Amstrad.

Specific features of Kuma Forth rarely found on other implementations are:

- Floating Point
- String facilities
- Real Time Clock
- Function Key Suport
- Genuine cursor controlled Screen Editor
- Tape based Virtual Memory System
- Stack Printing Utility
- Primitive Word Generation
- Fully compatible with ROMs & RSXs

A comprehensive, 120 - page manual is supplied with the cassette which includes an introduction to Forth, programming techniques, extensions and a glossary of words used in Forth.

Jon Day, Sales Manager of Kuma said: "Our version of FIG Forth has been fully implemented on the Amstrad, taking maximum advantage of the facilities offered by both the language and the micro. It brings all the benefits of this powerful language within the reach of every Amstrad user, whether they are experienced or novice Forth programmers".

Kuma Fig Forth for the Amstrad costs £19.95 including VAT and is available from all good microcomputer stores. For further details of this and the full range of Amstrad software from Kuma contact Jon Day, Kuma Computers Ltd, 12 Horseshoe Park Pangbourne, RG8 7JW. Tel. 07357 4335. □

THE ATARI 130XE — 128K FOR £170

Atari's new machine — the 130XE — is cheap, and there's no denying it.

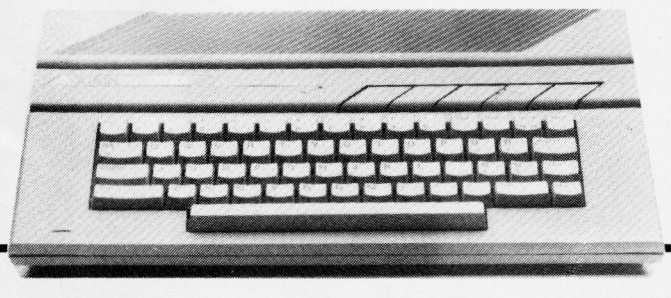
For a machine retailing at just under £170, the 130XE has an impressive specification and offers full compatibility with the Atari 400, 800, 600XL and 800XL micros.

Although the 130XE's 6502c processor is clocked at a moderate 1.79MHz, the machine is capable of displaying 256 colours simultaneously, and offers 11 different display modes although display resolution is a conservative 390 X 192 pixels.

Other features include:

- Serial I/O port
- Composite video output
- Cartridge slot
- 2 on-board A to D convertors
- Inbuilt disk interface
- 5 text display modes

All Atari peripherals are compatible with the 130XE.
For further details contact Angela Oakes/Roger Cox on 01 493 0000/821 5555. □



DISK-DOUBLER COULD BE A TROUBLER

MGA Microsystems of Tenterden in Kent, have come up with a neat gadget that cuts Write-Enable notches into 5.25-inch disks, permitting the reverse side of single-sided floppies to be used.

Retailing at a conservative £4.95 excluding VAT, the Disk-Doubler is reckoned to be the first device of its kind available here in the UK. Similar tools have been around for some time in the USA, but as Jon Gurr of MGA points-out, prices are high: "We've been looking at disk-notchers for five years, and the only problem was cost. Even now, companies in the States are asking between fifteen and twenty dollars!"

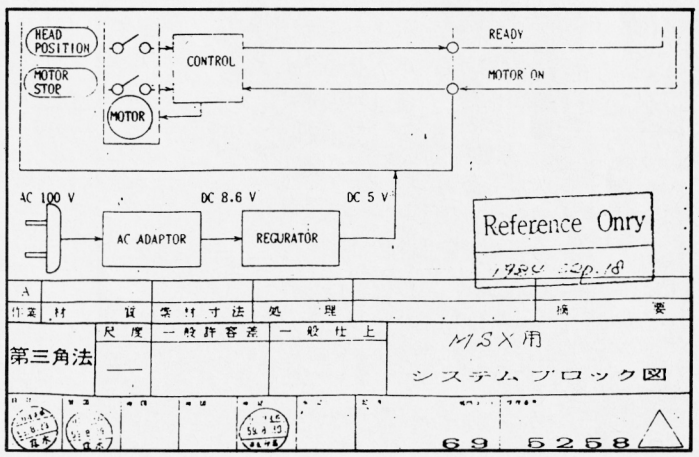
The Disk-Doubler is said to offer 'significant savings' by effectively doubling the capacity of 'notched' disks. Do note, through, that single sided disks are so named because the reverse side is of dubious quality, and is not covered by guarantee. Astute *Computing Today* readers might argue that the veracity of diskette surfaces can be tested. However, I would counter-argue that wise distrust is the parent of security.

With proverbs like these, who needs statistics?

MGA Microsystems is at 140 High Street, Tenterden, Kent TN30 6HT. Tel. 05806 4278.
JC □

SHURELY SHOME MISHTAKE!

We at *Computing Today* are well aware that People Who Live In Glass Houses Shouldn't Throw Stones, but this month's *Grauniad* award for good spelling must surely go to the unknown Yapanese originator of the diagram below. 'Regurator' is *almost* but not quite permissible, but 'Reference Onry'? On an official stamp?? Really!
JC □



NO CUSTOMS DUTY ON SOFTWARE

HM Customs and Excise have announced that, by virtue of a European Council resolution, imports of computer 'carrier media' (eg magnetic tapes and disks) can be valued, for customs duty purposes, as if they were unrecorded.

The cost or value of the data or information on the tape or disk may be ignored. This change, which came into effect on the 1st of May 1985, does not apply to valuation for VAT.

An amendment to the Customs notice on software and media, Customs Notice 252, will be issued in due course. Meanwhile, traders who require further information should contact HM Customs and Excise Valuation Branch, Vintry House, Queen Street Place, London EC4R 1QB. □

MOBILE CAREERS ADVICE CENTRE

People wanting a job in computing will now be able to learn about the industry at first hand.

Computeach International Ltd, the computer training specialists, have introduced a new mobile careers advice centre to tour the country.

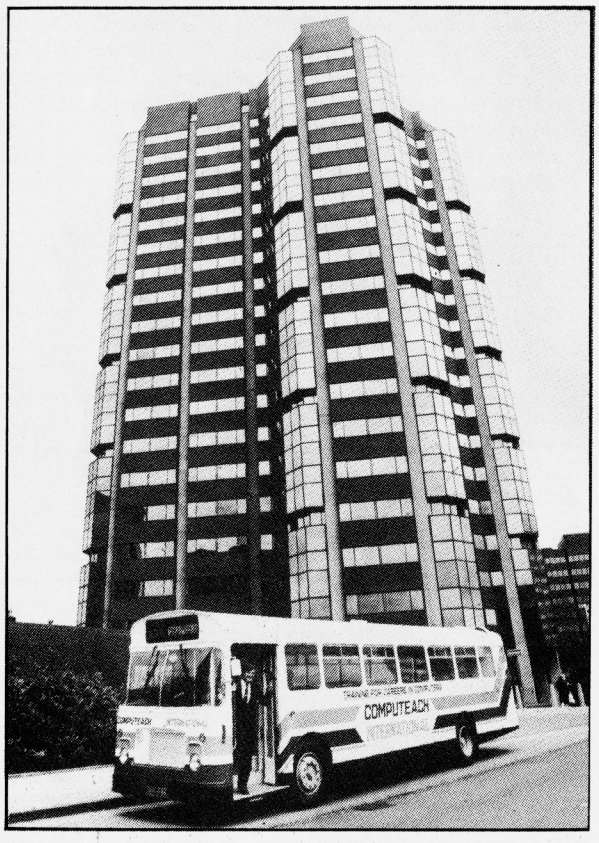
The former Midland Red single decker bus has been completely re-fitted with video and computer equipment together with tables and seating, and a team of Computeach careers and computer training representatives will accompany the mobile advice centre throughout the United Kingdom. They will be calling at main towns and cities to advise people about careers in computing together with computer training courses.

The bus is expected to start its tour in the next few weeks in the South West of England. Roger Walton, group marketing director said: "People prefer to see at first hand what computers and computing is about.

"By taking our advice centre on the road we can help dispel many of the myths that exist about careers in computing".

Further information from:

Computeach International Ltd,
8th Floor,
Metropolitan House,
1 Hagley House,
Five Ways,
Birmingham,
B16 8TG. □



TEACHING WITH TATUNG

If you have tried writing educational programs for your school or for the general market, you may have found that standard programming languages such as BASIC are unsuitable for presenting large quantities of information, or for assessing answers in plain English.

Microtext is a package designed specifically for the development of Computer Assisted Learning (CAL) programs and a Version of Microtext for the Tatung Einstein has been written by Transdata Limited enabling Einstein users to produce educational programs simply and quickly.

Microtext has a full screen colour editor for input and alteration of screens of information. The flow between, and the presentation of the screens is controlled by a set of simple commands. Special Response Matching facilities are provided to enable answers to be processed, and to determine whether or not the correct answer has been given. Reports on the students' progress and scores can be written to files for later assessment.

Existing microtext implementations for the BBC 'B', CP/M 80, CP/M 86, and soon to be announced MS-DOS version, mean that programs written in Microtext will run on a large number of popular micros.

Microtext for the Tatung Einstein cost £47.00 and is available from Transdata Limited. Tel. 0705 49556 □

NCC HELPS YOU REGISTER UNDER DATA PROTECTION ACT

Registration under the Data Protection Act (1984) will be required of all organisations holding and automatically processing personal data — data relating to living individuals — during a six month period starting in the autumn of 1985.

Failure to register is an offence, leaving an organisation open to an unlimited fine. What do you have to do? What are the key decisions to be taken? What help do you need to complete the registration forms? What are the pitfalls to watch out for? What follow-up measures will be needed? To help organisations deal with these problems, the National Computing Centre (NNC) will be running a series of seminars in locations throughout the country during the period 9th-20th September.

NCC specialists in data protection have been consulting with the Government and industry at all stages leading to the passing of the Act. Subsequently they have been in close contact with the Registrar (ex-NCC Deputy Director, Eric Howe), and with numerous organisations which have taken the lead in working on the practical implementation of the Act. The knowledge arising from all this work will be available to seminar attendees.

The NCC believes that the Act represents an important step forward in the protection of data concerning individuals and that compliance with the Act will not be too burdensome for those who approach the matter systematically. The seminars will offer practical guidance to organisations at this critical stage in responding to the requirements of the Act.

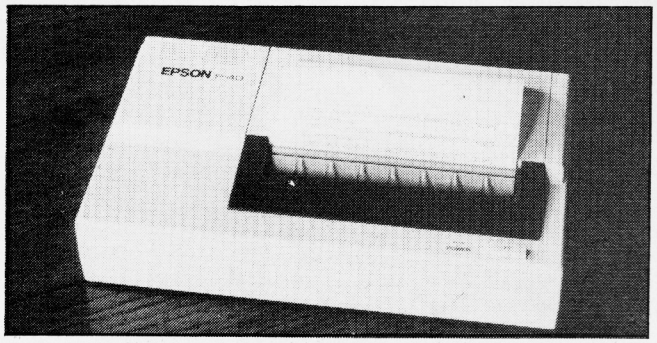
For further information contact the Seminar Administrator on 061-228-6333. □

EPSON PORTABLE PRINTER UNDER £100

New from Epson is the P-40, an 80 column thermal dot-matrix, battery or mains operated printer.

The P-40 is available with either Centronics or RS232 interface, with a print speed of 4cps. Included in the price is a starter pack containing a free mains adaptor, an operation and configuration guide, and sample programs for most micros.

● For more details contact VSI Electonics, Roydonbury Industrial Park, Horsecroft Road, Harlow, Essex CM19 5BY. Tel. 0279 28666. □



FIRST EUROPEAN PERSONAL ROBOTICS CONFERENCE AND EXHIBITION

The first European Personal Robot Congress is being held in London at the beginning of July.

Co-sponsored by the IEE, British Personal Robot Manufacturers Group and *Practical Robotics*, the three day event is to consist of a Conference looking at the personal robot market, technical developments and the industry in general. Speakers from Europe and the USA complement some of Britain's foremost experts in this area.

Running in parallel with the conference are a series of specialist workshops and an exhibition showing the best in personal robotics. The UK finals of the Micromouse competition and final heats of the World Robot Ping-Pong competition will be held at the same time.

The fee for the conference is £250 per delegate plus VAT, which includes all conference documentation, technical sessions, exhibition, refreshments, pre-luncheon drinks and luncheon.

For further information contact:

Miss Louise Marriot
Oyez Scientific and Technical Services Ltd
Bath House
3rd Floor
56 Holborn Viaduct
LONDON
EC1A 2EX
Telephone: 01-236 4080
Telex: 888870

NEW PERIPHERALS FOR ENTERPRISE

Enterprise Computers have announced a range of new peripherals and cables for the Enterprise 64 micro.

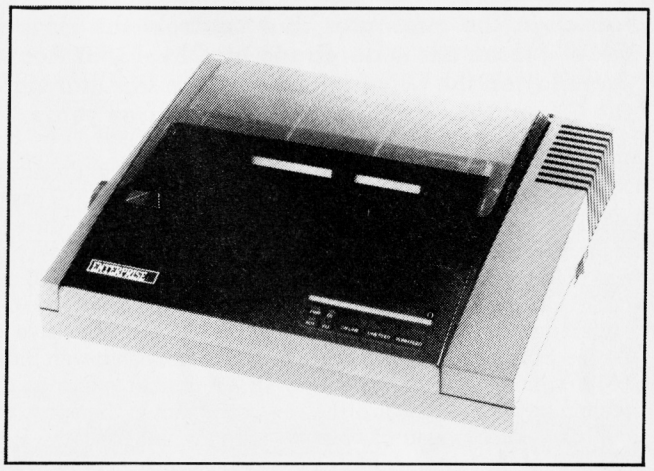
The peripherals are the EP80+ dot matrix printer (£239.95), a Colour monitor (£349.95), and a Joystick interface (£9.95).

The printer is manufactured under the Enterprise label by Mannesman Tally, and has a built-in 2K RAM pack capable of downloading user-defined characters. The resident character set offers a range of printing styles and will print high-resolution graphics.

The monitor, also under the Enterprise label, is manufactured by Microvitec. It has a dot pitch of 0.43mm, tube resolution of 653 x 585 pixels and allows full use of the 80 column text mode. The Enterprise monitor also includes an on board speaker and the ability to transfer the computer sound output to an external amplifier.

The joystick interface plugs directly into the Enterprise control ports, and permits the use of any standard 9 pin D ring interface joystick.

Enterprise have also produced a colour/sound monitor cable, a parallel printer cable and a serial/networking cable.



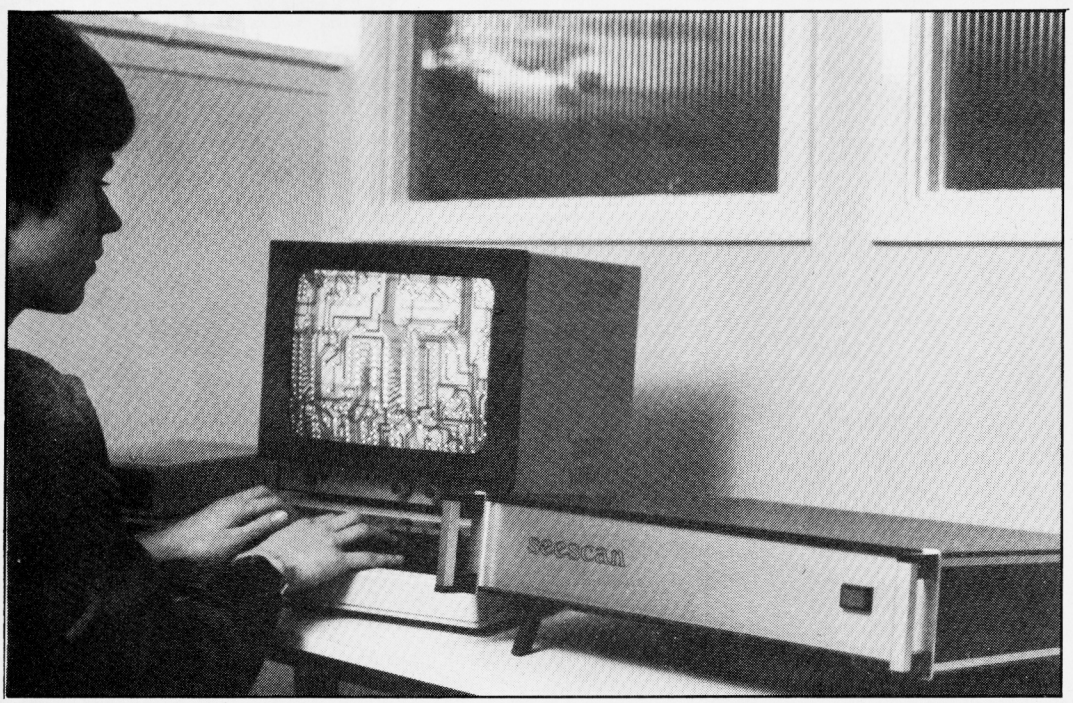
SEESCAN ANNOUNCE NEW IMAGE PROCESSING SYSTEMS

Seescan, a design and consultancy company specialising in image processing systems have added two new systems to their range.

The RD2566 and RD2567 incorporate a 256 x 256 x 6 bit (RD2566) or 7 bit (RD2567) framestore with an on-board Z80 processor running at 6 MHz (8 MHz optional). The systems have RS232 and BBC 1MHz bus interfaces for terminals, powerful graphics facilities and an extensive software development package which allows users to develop software easily to customise their system. The graphics facilities include user definable overlays and look-up tables which can be programmed to perform contrast enhancement and gamma correction. Both the RD2566 and RD2567 have eight bit video outputs allowing the user to display 256 shades of grey. It is possible to expand the RD systems by adding three extra frames of image memory.

The software supplied with the system includes a Z80 monitor and ZASSM — a Z80 assembler. The Z80 monitor is accessed via a terminal connected to the RS232 port or via the 1MHz bus of the BBC Microcomputer. It includes full screen memory editing facilities, copy and fill commands as well as an in-line assembler. ZASSM compiles Z80 assembler files written under a suitable word processor. Compiled files can be downloaded into framestore User RAM and executed.

Seescan was formed by research fellows of Cambridge University to market their expertise in image processing. Besides the RD systems, Seescan also manufacture a very powerful high resolution monochrome/colour image processing system. As a service to customers, Seecan offer an applications software design capability for the RD systems.



● For further information contact Dr Philip Gaffney on 0223 314553 / 61376.

AUCTION TO AID FAMINE RELIEF

Software donated by leading software houses is to go under the hammer at an auction in support of Ethiopian famine victims.

The auction is to take place at the headquarters of the Greater London Council, County Hall, on Saturday June 15th, comencing at 11.00am.

Old and unwanted software is sought by the organisers of the event, Soft Aid and Computer Trade Weekly, and all software should be sent to:

Specialist Retail Press Ltd.
Business Technology Centre
Bessemer Drive
Stevenage
Hets SG1 2DX
Tel. (0438) 316561

JC □

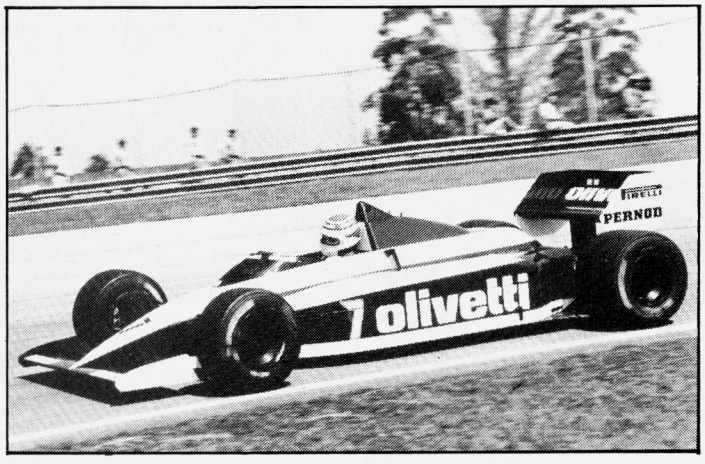
OLIVETTI/BRABHAM FORMULA 1 PARTNERSHIP

Although the computer that controls the Brabham-BMW's team car is not made by Olivetti, or Acorn for that matter, the Olivetti logo will be a familiar sight on the Grand Prix circuit over the next three years.

Olivetti is backing the Brabham drivers Nelson Piquet and Francois Hesnault in their challenge to the Maclaren team's domination of Formula 1.

Behind the scenes, Olivetti engineers are working with Bernie Ecclestone's Brabham-BMW team to apply computer technology to the solution of the complex problems that surround the development of a winning Formula 1 car.

Vittorio Levi, Olivetti's Executive Vice President for Operations said: "Formula 1 racing appeals to an ever increasing number of people, and one in which the high technology in which Olivetti specialises is a decisive factor. Our partnership with Brabham BMW further expands our involvement in the sport; we are already responsible for processing results of both qualifying rounds and Formula 1 races". □



LORDS COMMITTEE APPROVE ESPRIT WITH RESERVATIONS

In a report recently published, the House of Lords Select Committee on the European Communities approves ESPRIT (European Strategic Programme for research and development in Information Technology) as a valuable initiative deserving support, but the Committee say that some of Esprit's assumptions and certain aspects of the way it is run need to be re-examined. And they warn against unrealistic expectations of what Esprite alone can achieve.

The Committee say that points in ESPRIT's favour include:

- it is the first attempt to set up a complex but vital EEC-wide collaborative project involving the private sector.
- it has proved a catalyst for important collaborative joint ventures between Euro-firms more used to competing against each other.
- it may help close the technolgy gap between Europe and Japan and America;
- it may win back for Euro-manufacturers a sizeable part of the European market for Information Technology products.

The Committee's criticisms of ESPRIT include:

- too little thought has been given to exploiting ESPRIT's researches and to producing products for the market place
- there is duplication between ESPRIT and similar national programmes like the UK's Alvey programme
- ESPRIT project assessment is poor; too academic and not commercial enough
- small firms are not getting enough encouragement or a fair share of Esprit funds
- ESPRIT is taking limited EEC research funds away from other important research areas such as biotechnology
- ESPRIT is not putting enough stress on software
- ESPRIT monitoring is poor. It lacks a "presiding genius" to pull its miscellaneous projects together.

The Committee say that Esprit cannot succeed without;

(a) a genuine Euro common market which at present rates of progress is a long way off.

(b) policies to tackle the serious shortages of skilled manpower qualified to work with the new technologies.

(c) large-scale support for commercial product development. □

YOUNG ENTHUSIASTS GET BCS BACKING

The Council of the British Computer Society has launched an Association of Young Computer Enthusiasts aimed at providing an educational and ethical framework for computing activities by people in the 16 to 18 years agegroup.

The Council's move results from recommendations made at the Members' Congress at Reading University in March this year, and comes partly as a consequence of the Royal Charter, granted in July 1984, which implies public responsibility on behalf of the Charter holders.

Further details from:

Mr D W Harding
Secretary General
The British Computer Society
13 Mansfield Street
London W1M OBP
Tel. (01) 637 0471

JC □

TIMBER-SAVING PAPER FOR ROUTINE LISTING

Available from Action Computer Supplies is a new addition to the range of continuous listing paper, Checklist, manufactured by Standard Check Book.

Checklist is made by a thermo-mechanical process which produces strong, low-cost paper with good opacity, ideal for routine listings.

"The thermo-mechanical process not only produces good, economical paper, but helps with timber conservation" says Action Sales Director Dick Sheppard. "TMP, as it's called, utilises about 96 per cent of each tree for paper, as against less than 60 per cent for the standard chemical process".

Price per 2,000 sets of plain, one-part, 11in x 9.50in is £12.90 and 11in x 14.50in, £16.10.

Action Computer Supplies are at, 6 Abercorn Trading Estate, Manor Farm Road, Wembley, Middlesex HAO 1WL. Tel: 01-903 3921.

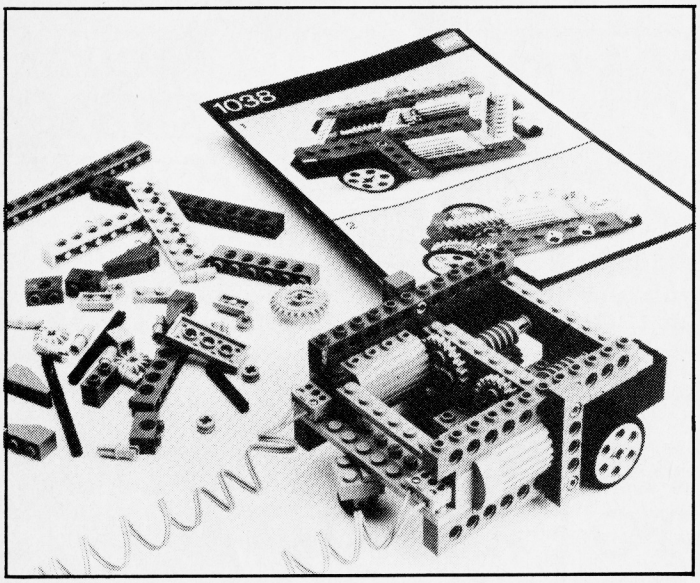
LEGO UNIVERSAL BUGGY FROM GRIFFIN

Suppliers of Lego Technic Kits, Griffin and George, have added the outstanding Lego Universal Buggy to their range.

As a buildd up kit it is ideal for computer control learning but the adaptability of Lego will give the buggy a wider appeal among technology teachers who will find the design-and-build capability of Lego in tune with todays teaching methods.

Price £25.95.

● Further details from Nick Swift on 01 997 3344 Ext 311.



CPU OFFER DAISY-WHEEL FOR UNDER £250

Despite the EEC's recent tariff increases on imported typewriters and printers, CPU Peripherals Limited has released a high quality, professional daisywheel printer for a retail price of just £249.

Designed primarily for word-processing applications, the new Uchida DWX-305, priced at £249.00 is undoubtably the most competitive model in its category of high quality daisy-wheel printers. Companies or retailers ordering several units will be eligible for CPU Peripherals' quantity discount scheme.

Operating at a print speed of twenty characters per second, the specifications of the Uchida DWX-305 are on a par with more expensive models currently available. The printer uses the industry standard Qume daisywheel, a multi-strike ribbon cartridge, plus Centronics and RS232C interfaces, making it compatible with a wide range of micros. In addition, the DWX-305 features ten or twelve character pitch selection, a 1/20 inch horizontal increment for accurate shadow printing, as well as dot graphics.

Other features include: 13 inch plated length; ¼ inch minimum line feed pitch; pin, and particularly for correspondence applications, an automatic paper feed; a copy capability of original plus three copies; automatic eight level impression control and 256 character memory buffer.

Further information from:

CPU LSI Computers PLC
Copse Road
St Johns
Woking
Surrey GU21 1SX
Tel. 04862 73883. □

FULL PRESTEL FOR ATARI USERS

Full PRESTEL facilities for ATARI users — that's the welcome news from modem manufacturers MIRACLE TECHNOLOGY (UK) LTD.

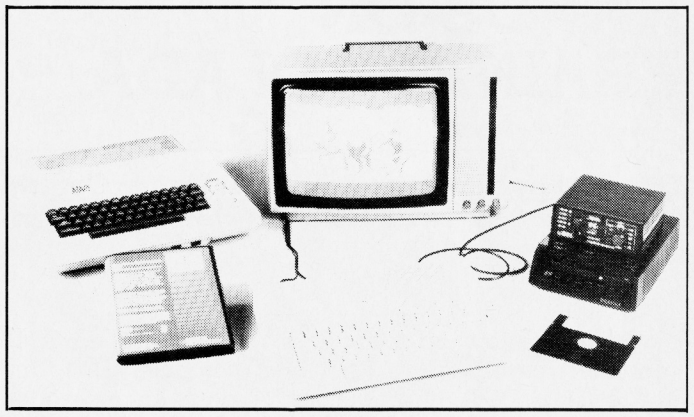
For the first time, with their Multi-Viewterm/Datatari modem interface and software package, all PRESTEL facilities are available to the Atari owner who up until now has had to make do with the very limited access given by the 850 interface.

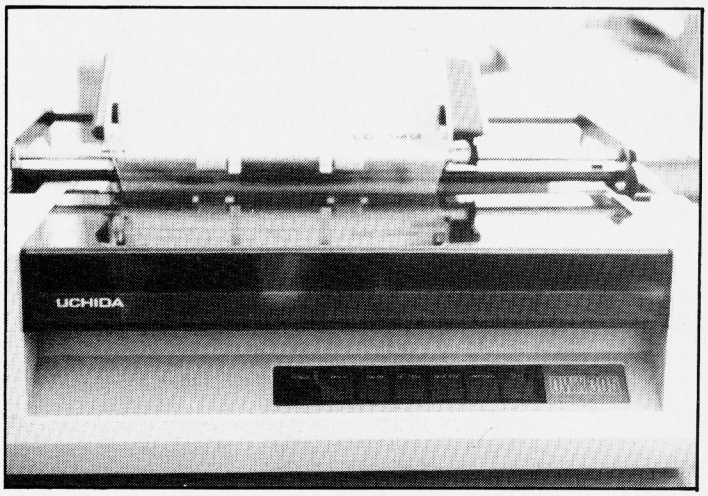
Now the 850 interface — restrictive because of its inability to handle split baud rates — is no longer necessary. The Multi-Viewterm/Datatari package allows baud rates of 1200/75, 75/1200, 1200/1200 and 300/300, and includes both ATARI 13-way peripheral port plug and standard 25 way plug to fit the Miracle Technology World Standard modems — and a large number of other modems too.

The Datatari interface and Multi-Viewterm disk-based software give the Atari owner not only access to PRESTEL and other previously inaccessible Viewdata systems, but also to Electronic Mail, Telex, Database and User-User communications — and allow Telesoftware Downloading.

The package can be used with ATARI models 400, 800, 600XL, 800XL, 65XE, 65XEM, 65XEP, and 130XE.

Multi-Viewterm/Datatari will be stocked by a large number of Atari main dealers including the biggest hardware and software suppliers to the UK Atari community, Silica Shop and Software Express. By mail order, the package costs £59.95 inc VAT (+£1.15 UK P/P) from Miracle Technology (UK) Ltd., St Peters Street, Ipswich IP1 1XB. Tel 0473 - 50304.





COURSES AND TRAINING

Education around the U.K.

HANDS-ON TRAINING IN ADA

This summer, the Microcomputer Unit in Central London launches a new five-day workshop — Programming with Ada.

The first on-site course is due to run in early August. As with all workshops organised by the Microcomputer Unit, the emphasis is placed on the practical approach. Each attendee will have his/her own terminal for extensive hands-on exercises in writing, compiling and executing programs in Ada. This is the only way to gain a real understanding of this important language.

Sponsored by the United States Department of Defense, the Ada language was designed to solve the programming problems associated with the development of large, reliable, real-time systems with a long lifespan. It is now one of the three approved languages in which all Department of Defense software must be written and this strategy is being followed by both NATO and the Ministry of Defence. In addition, it is expected to be used in non-military applications as this was one of its design requirements. Admittedly, the lack of Ada compilers has delayed its arrival on the market. This is due to the stringent validation requirements imposed by the Department of Defense. However, this will ensure an extremely reliable code which is highly portable and this can only result in the eventual success of the language.

The Microcomputer Unit is preparing early to lead the way with training in Ada programming. For a detailed syllabus on this or other courses contact: the Microcomputer Unit, 235 High Holbom, London WC1V 7DN, tel. 01-405 3020, telex 25964.

AN INTRODUCTION TO PASCAL PROGRAMMING

The City of London Polytechnic is offering a three-day, full-time course in July 1985 in PASCAL programming which starts at an introductory level and emphasises the features of PASCAL that set it aside from such languages as BASIC, FORTRAN and COBOL.

The course will be of particular interest to home micro users, school teachers and business people looking for familiarity with a modern computer language.

The course takes place in London on 16, 17 and 18 July and the fee is £60.00 per person. Further information and application forms are available from the Short Course Unit at the Polytechnic, 84 Moorgate, London EC2. Tel: 01-283 1030 (extension 324).

'C' WORKSHOP

The best way to learn a new language is to use it under the guidance of an expert instructor. That is just what happens in a new short course entitled. "PROGRAMMING IN C: A HANDS-ON WORKSHOP".

This 4-day course provides each pair of participants access to a UNIX system with C compiler and library use in learning to write, compile, and execute C programs; make full use of C's powerful structured programming features; implement C programs for real-time applications; write system I/O drivers using C; link assembly language modules to C programs; develop programs using C library functions; and use the UNIX "Programmer's Workbench" for C development.

Priced at £635 + VAT, this course is held throughout Western Europe.

Further details: (0372) 379211

UNIX WORKSHOP FROM INTEGRATED COMPUTER SYSTEMS

A solid foundation in the use and facilities of the UNIX operating system is provided in this four day course, entitled "UNIX: A HANDS-ON INTRODUCTION".

Throughout the course, extensive hands-on experience using the UNIX operating system reinforces key concepts and applications while providing a coherent, in-depth introduction to UNIX and its most important facilities.

Each pair of participants is provided with a terminal for use in class. Exercises include: using shellscripts in applications; executing programs under UNIX; employing UNIX piping and filter facilities; using files and directories; and sending and receiving electronic mail.

Priced at £635 + VAT, this course is held throughout Western Europe.

For more information contact Adrian Nettleship on (0372) 379211.

MAN MACHINE INTERFACE

A comprehensive introduction to the principles of human factors engineering and the application of these principles to the hardware and software design of effective man/machine interfaces is offered in a short course from Integrated Computer Systems.

This 4-day course, entitled "DESIGNING EFFECTIVE MAN/MACHINE INTERFACES", combines workshops and case studies with lecture presentations to provide an in-depth understanding of key concepts and techniques along with the concrete guidelines for their implementation. Case studies and examples include: screen layout and design; improving man/machine dialogue; selecting and configuring alternative input & display devices; evaluating graphics displays; and video tape demonstrations illustrating interface techniques.

Priced at £585 + VAT, this of course is held throughout Western Europe.

Phone (0372) 379211

ARTICLE SUBMISSIONS

Computing Today welcomes articles submitted by readers, but potential contributors are asked to observe the following guidelines:

- Feature articles should be typewritten with double line-spacing if possible, and should ideally cover between 8 and 16 sides of A4.
- Program listings when absolutely necessary should be kept short. When listings are supplied they must be accompanied by text describing the algorithms employed and, if possible, a flowchart.

Articles should be sent, along with an accompanying letter describing the nature of the article, to:

SUBMISSIONS
COMPUTING TODAY
1 Golden Square
London W1R 3AB.

Please ensure that your name and address is printed clearly on your covering letter.

READERS

HELP US TO HELP YOU:

MENTION

COMPUTING TODAY

WHEN RESPONDING TO ADVERTISEMENTS

HISOFT

High Quality Microcomputer Software

We are delighted to announce that both **HiSoft C** and **HiSoft Devpac** have been nominated in the Home Software category of the British Microcomputing Awards.

These nominations are particularly pleasing since they mean that the computer industry is recognising what our customers have been saying for years: HiSoft produces high quality utility and language software at realistic prices with full technical support, a rare bonus in the home software jungle.

We have been crafting programming tools for many years and we can now offer an unrivalled set of development packages on the popular home microcomputers. The Product Availability Table below shows our current range together with UK inclusive prices; please feel free to write for technical details of any product or for export information. Our friendly technical and sales team are always happy to answer any telephone enquiries on (0582) 696421.

Product Price Availability Table

Product / Machine	Pascal	Devpac	C	Ultrakit	Font 464
ZX Spectrum	25-00	14-00	25-00	9-45	
Amstrad CPC464	29-95	21-95	*		7-95
MSX	29-95	19-95			
CP/M-80	39-95	39-95	*		
Sharp	39-95	25-00			
Sinclair QL		19-95 (MON QL)			

*Available soon. All prices in £ sterling.

180 High Street North, Dunstable, LU6 1AT. (0582) 696421

BUSINESS COMPUTERS
Epson PXB £900 (£872) £892. Commodore PC 10 £1595 (£1564) £1664. Sanyo MBC 775 £1920 (£1899) £1999. Cannon A200C £1609 (£1586) £1686. Sanyo MCB550 £723 (£699) £799.

ORIC AND SINCLAIR COMPUTERS
Oric Atmos computer 48K £99 (£99) £109. MPC40 Oric printer/plotter £109 (£110) £122. Sinclair pocket TV £97 (£95) £101. Sinclair QL Computer £379 (£378) £399. QL Floppy disc interface £107 (£103) £109. Sinclair Spectrum Plus Computer 48K £127 (£129) £149. Kit to upgrade the Spectrum to Spectrum Plus £30 (£30) £40. Microdrive £40 (£50) £60. RS232 interface 1 £49 (£50) £60. Special offer:- Microdrive + Interface 1 + 4 cartridges £97 (£99) £107. Blank microdrive cartridges £2-50 (£3) £4. Spectrum floppy disc interface (See Cumana disc section for suitable disc drives) £97 (£89) £99. Interface 2 £20-45 (£20) £24. 32K memory upgrade kit for 16K spectrum (issue 2 and 3 only) £31 (£28) £30. Spectrum Centronics printer interface £46 (£42) £47. ZX printer has been replaced by the Alphacom 32 £71 (£69) £82. 5 printer rolls (State whether Sinclair or Alphacom) £13 (£16) £21. ZX81 computer £35 (£35) £45. 16K ram packs for ZX81 £28 (£25) £30.

COMMODORE COMPUTERS
Commodore C16 Starter Pack £119 (£119) £151. Commodore Plus/4 £147 (£142) £174. Commodore 64 £189 (£194) £226. Convertor to allow most ordinary mono cassette recorders to be used with the Vic 20 and the Commodore 64 £9-78 (£9) £11. Commodore cassette recorder £43 (£44) £50. Centronics printer interface for Vic 20 and the Commodore 64 £45 (£41) £46. Disc drive £197 (£201) £232. 1520 printer/plotter £99 (£96) £111. MPS801 Printer £189 (£191) £229.

AMSTRAD, ATARI AND ENTERPRISE COMPUTERS
Amstrad Colour Computer £342 (£348) £399. Amstrad Green Computer £232 (£247) £287. Atari 130XE computer £158 (£163) £183. Atari 520ST computer £634 (£616) £656. Atari 800XL computer + recorder £120 (£123) £143. Atari 800XL Computer + disc drive £229 (£230) £260. Atari data recorder £34 (£37) £47. Atari disc drive £172 (£171) £191. Atari 1020 printer £93 (£99) £115. Enterprise 64 computer £234 (£236) £256.

ACORN COMPUTERS
Acorn Electron £119 (£119) £139. BBC Model B £354 (£343) £383. BBC Model B with disc interface £474 (£461) £491. Colour monitor £188 (£228) £268. Kenda double density disk interface system £137 (£131) £141. See below for suitable disc drives.

CUMANA DISC DRIVES
To suit disc interfaces of Sinclair QL, Spectrum, and BBC B. Single:- 40 track single sided £164 (£163) £183, 40 tr double sided £169 (£168) £188, 80 tr ds £219 (£209) £229. Dual:- 40 tr ss £294 (£280) £320, 40 tr ds £325 (£325) £365, 80 tr ds £414 (£390) £430.

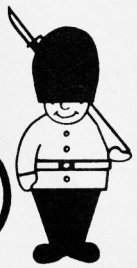
PRINTERS
MCP40 4 Colour printer/plotter £109 (£110) £122. Brother HR5 £162 (£161) £193. Shinwa CTI CPA80:- centronics parallel version £218 (£222) £258, RS232 version £238 (£240) £282. Cannon PW108A £312 (£308) £358. Epson RX80 £249 (£249) £282. Epson RX80F/T+ £283 (£281) £316. Epson FX80 £339 (£305) £335. Combined matrix printers and electric typewriters:- Brother EP22 £135 (£124) £144, Brother EP44 £230 (£226) £246.

SWANLEY ELECTRONICS

Dept CT, 32 Goldsel Road, Swanley, Kent BR8 8EZ, England.
TEL: Swanley (0322) 64851

Official orders welcome. All prices are inclusive. UK prices are shown first and include post and VAT. The second price in brackets is for export customers in Europe and includes insured air mail postage. The third price is for export customers outside Europe (include Australia etc) and includes insured airmail postage.

MICRO GUARD



COMPUTER POWER PROTECTION

Uninterruptible Power Supplies
Constant Voltage Transformers
Emergency Power Units

Made in England
Comprehensive 12 month warranty
Next day delivery service

24 Foregate Street,
Worcester WR1 1DN.
Telephone: (0905) 21541
Telex: 339692

Computer Programmers, Systems Analysts.

They're asking for you from Africa to the Pacific.

Voluntary Service Overseas is looking for computer programmers and systems analysts, to work in and help train third world communities.

Applicants should be aged between 23 and 65, without dependants and willing to accept no more than the 'local' rate of pay.

If you have the right qualities and experience and you're free to go, please believe you're needed urgently!

(For more information, complete and return the coupon now.)

VSO

VOLUNTARY SERVICE OVERSEAS

Why not?

I'm interested in volunteering. My qualifications or experience are:

_____

_____

Name _____

_____

Address _____

_____

CT/6/85

Post to: Enquiries Unit, Voluntary Service Overseas, 9 Belgrave Square, London SW1X 8PW. (22p S.A.E. appreciated) Charity no. 313757.

TALKING SHOP

Don Thomasson

At question time in a Computer Club meeting or a session in the local library, someone is bound to ask 'What do you use your computer for?' The question always makes me sigh, because the answer depends so much on personal tastes and interests. My usual response begins; 'I use it to do things which I couldn't do in any other way.'

No. That isn't an evasion. It's a starting point, which I can illustrate with examples.

For instance, I sometimes have a need to solve power series expressions. That may be possible on a manual basis, but it becomes a lot easier when you can call a computer to help. The user has to contribute, by writing a suitable program, but that is usually a minor task.

I like to play two-handed card games, and the computer will satisfy that whim. I'd like to have it play three hands at bridge, but I haven't managed to work out a suitable program yet.

Lightweight applications? Yes, they are, but they can be important to me, though perhaps not an adequate justification for buying a computer that would be used for nothing else.

'You write a lot. I suppose you use a word processor?'

Er, no... I don't, though I may have to before long. A lot of people are suprised at my attitude, which stems from the nature of much of my writing work. I find it difficult to maintain proper continuity when I can only look at twenty-five text lines at a time. True, I can scan backwards and forwards to see more text, but that isn't the same as having several sheets of typescript spread out for direct reference. I have the same problem with screen listings. It's much easier to work from a printout.

I am now being presed to use a word processor to save the need to retype everthing afresh. The first snag concerned compatibility. The best word processor I have access to is Spellbinder on a Sorcerer, but that produces hard-sectored discs that would be unusable for most systems. I need to pick a machine that will be compatable with the typesetting system, but which pleases me in other respects. It would be absurd to saddle myself with two machines, one for word processing and another for everything else.

Mind you, I do use small database systems, particularly in preparing indexes for books and the like, but that tends to be rather specialised. As a hobby, I maintain a card index of Grand Prix drivers and their results. It runs to more than five hundred cards, and each card may cover up to 16 racing years and 176 races. (Until someone betters Graham Hill's record, at least.) Each year involves up to sixteen results, and there may be two entries per year, because a driver changed teams mid-season. Now, this I would love to automate, so that I could extract statistics at will, but tentative studies have produced discouraging results. I'm not not buying a mainframe to do the job, so I'll have to go on trying various compression schemes until I can fit the data into a reasonable space.

□□□

SO MUCH for my personal likes and dislikes. Those of the middle generation of the family are not too dissimilar. There are lists of chart-busters to be maintained, and every now and then a new program surfaces. It may be a game, or a dungeon-designer for dragons, or a disc doctor to tidy up discs that have gone crook, but the main point is that it is a challenge, especially when written in assembler. It also makes a change from formal programming during working hours...

The youngest generation just likes to prod keys and watch the screen change, but she hasn't reached her first birthday yet!

The truth is that the computer is all things to all persons (as we have to say these days). What you do with it depends a lot on your personal interests. If computing is your only interest, then you can amuse yourself by delving into the inner workings of the system and work out improvements. That is the way many of our contributors start the projects they write about. I find the process particularly fascinating, especially when I find a bug that hasn't become generally known...

Computer users who are into hardware construction have a much wider scope, and can spend many happy hours building and using complex control systems, but people who talk of controlling their homes by computer are probably going to bite off more than they can chew. I don't like the idea of a general-purpose microcomputer being tied down to a 24-hours per-day task.

The one thing that hasn't really been mentioned is the use of the computers to make money. Where some owners see their machines as just an expensive toy, others see a source of income. A very large number of people pay for their computers by writing about them, while others do the social secretary a favour by producing personal letters to club members. One of our family computers used to go to meetings of a radio-controlled racing car club, where it tabulated results and produced result sheets and summaries. Some firms are only too glad to have their accounts tabulated. Others prefer to keep their transactions to themselves!

But, to go back to the begining, the real point is that there are things that you can do with computers that can't be done in any other way. The desire to do these things must exist, of course, even if it tends to be restrained by a belief that the task concerned is impossible.

It is perhaps rather sad that so many people find the computer nothing better to do than play games. Not that I have anything against games, as such, but I feel they have rather taken over the computer scene, and now that interest is apparently waning the dearth of alternatives may become painfully evident.

□□□

IF YOU are stuck for an idea, and back numbers of Computing Today fail to suggest an answer, make a list of things you would like to do, quite apart from computing as such, cross out the obvious no-nos, and you may well get the inspiration you are looking for.

Don't, on the other hand, ask other people how they use their computers, because their answers will probably be meaningless to you - as mine probably were. The answers you need must come from yourself, from your tastes, likes, dislikes - and, above all, the amount of patience you are prepared to contribute towards amount of patience you are prepared to contribute towards achieving the objective which you select.

5 YEARS BACK

July 1980

Perhaps the most important long-term feature of the July 1980 *Computing Today* was an advertisement for the 'New Acorn Atom', forerunner of the BBC micro. Otherwise, the computer world was coasting through the summer months, gathering strength for the busy winter season. The editorial pages were mainly occupied by listings for the TRS-80, Nascom, and PET, though a letter from a reader suggested that there had been too much emphasis on PET programs in recent issues.

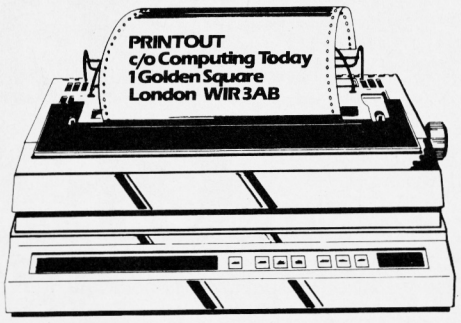
A sad note was struck by a user review of Compucolour, described as "... the colour graphics computer that never seemed to catch on". From the abysmal level of support described in the article, the failure to achieve popularity was scarcely suprising, though the writer did rather invite trouble, buying direct from New York. There was no user manual, discs had to be preformatted by external means, and the picture was jumpy. Then, a complete failure forced a reversion to the agents in this country, who at least loaned a replacement machine while the original was being repaired.

In one sense, repair sevices were more adequate in those days, in relation to the number of computers in private hands, but this state has not been maintained. Fortunately, the need for repair seems to be diminishing - once a computer is working, it stays working though 'infant mortality' remains rather high. Manufacturers have learned the hard way that proper design is important (the Compucolour seemed to have been grafted into a standard television set, with consequent problems!)

Incidentally, the printer of the day was the small Centronics, with 12cm wide metallized paper, offered half-price at £195 + VAT!

And the Newbrain was announced!

PRINTOUT



Your opportunity to ask questions, put us straight, seek advice.

PLACE FOR A JOURNEYMAN?

Dear Sir,
We wish to send one of our employees to the U.K. to study computing, develop his command of the English language and gain an insight into business administration, on a fee-paying basis. Period of course — up to two years.
Could you assist us by advising:-

a) suitable establishments covering these requirements, or
b) the appropriate body to approach for same?

If no accommodation is provided we would need the location to be accessible from the West London area.

Yours faithfully,
(C.J. Clarke)
Manager
Almansour Industries
Kilo 9 Medina Road,
P.O. Box 8413,
Jeddah,
Saudi Arabia.

A SATISFIED CUSTOMER

Dear Sir,
I like your approach to home computing and it is a pleasure to read a magazine which is not stuffed full of unreadable machine code listings.

It is rewarding to adapt one of your programs for, say, the BBC micro to my Spectrum, but at the same time the process is made more difficult because some of the commands are not relevant to the Spectrum, and I am not sure of their meanings. For instance, how do you convert the Beeb's 'VDU 4' into *Spectrumese*? What is the difference between MODE 4 and MODE 7?

Would it be possible to write a short article, or list, giving the major peculiarities for the common home micros? Much as I enjoy translating programs for other machines, time is usually limited. A simple conversion table would help greatly.

A final comment: what place does an advertisement on Vasectomy have in your magazine (February issue)? Is it to remind readers of the 'snip' they missed?

Yours faithfully,
J. B. Palmer
South Africa.

A list of the type you describe may be of benefit, and like the three men atop the deep shaft, we are looking into it.

HISOFT PSCAL — IN REPLY

Dear Sir,
In reply to the letter from David Link of Hisoft (*Computing Today* May 1985) I would like to inform him that the Pascal compiler reviewed was as stated in the review, and that the only program received was the compiler HP4D16.COM. No editor or sample programs were provided on the disk supplied. However, if Hisoft would like to supply me with the new version, or update my old, I would be very willing to review it, if *Computing Today* is willing to publish (*we are — Ed.*).

I would like to apologise to Hisoft for the inaccuracies that appeared in the review, but the information was correct at the time of writing.

Yours sincerely,
David Scott

CHRISTIAN MICRO USERS ASSOCIATION

Dear Sir,
We wold appreciate your mentioning the newly formed Christian Micro Users Association. We hope to link together a large number of Christian micro users and also to promote the use of micros in Church activities.

There is not only a great need to discover the few individuals and companies producing Christian software, but also to share the expertise and ideas of the many people who have sought to use micros in their church related activities.

Anyone who sends a large S.A.E. to.

Christian Micro Users Association,
c/o 6, Walkley Street,
Sheffield S6 3RG.

- will receive further details and a sample magazine.

Yours faithfully,
P.A. Clark (Secretary).

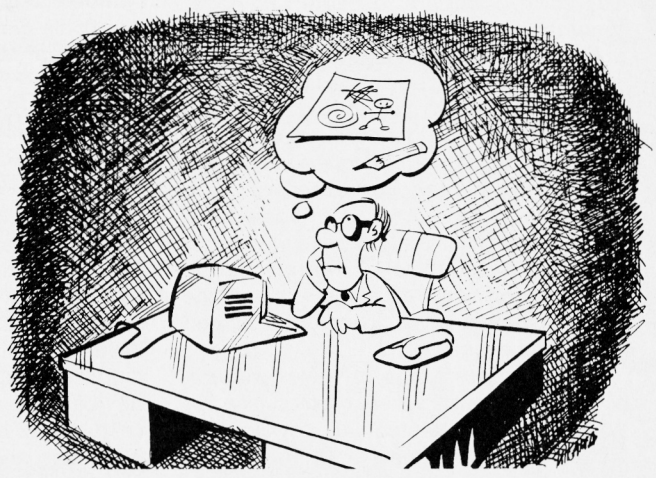
BBC PASSWORDS

Dear Sir,
I was impressed by the article BBC Passwords in your February issue. However, my knowledge of machine code is not great. I would like to use the programs described in the article on my CPC464, but the listings are for the BBC micro.

Could James Tyler or somebody please translate his two coding programs to run on my machine.

Respectfully yours,
Ivalson Soreno,
Auf der Weide.

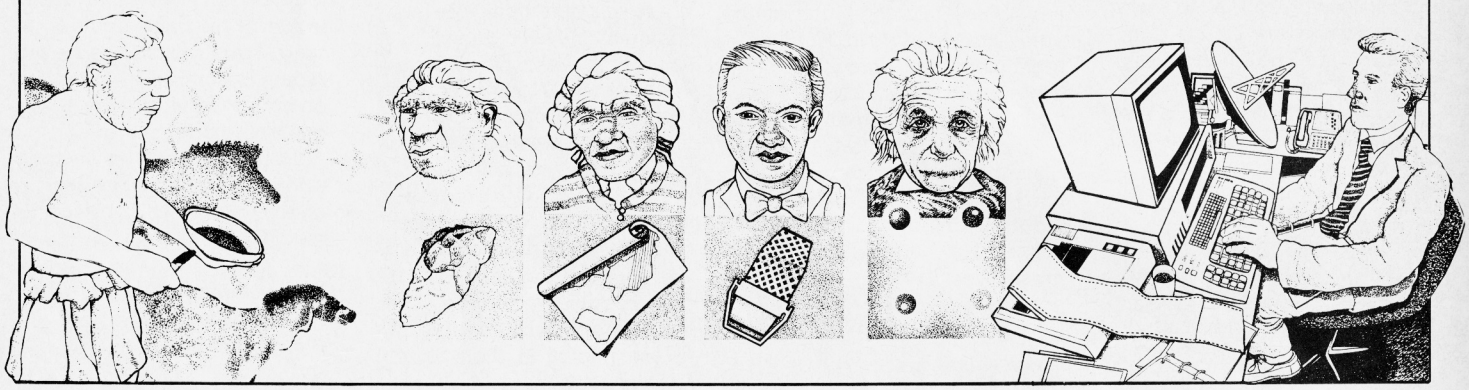
I'm afraid we can't help your there, Ivalson, but if any reader has developed a version of Passwords for the CPC464, we would be glad to pass it to you.



LESSONS OF HISTORY

Bill Horne

The late Fifties began to see the gradual introduction of transistor technology in computer design. Predictably, resistance to the 'new-fangled' devices was felt, and this lead to a curious situation in which computers based upon both valve and transistor technologies were being designed and manufactured.



Despite early pessimism, transistors did become a practicable proposition during the nineteen-fifties, and this brought about a major change in the nature of computing equipment. To understand the nature of this change properly, it is necessary to examine some of the basic characteristics of electronic circuitry.

The fundamental elements used in electronics are of two types: static or, inert elements will pass signals onwards, but at reduced power levels. To make up for this power loss, active elements are required. For many years, these were normally thermionic valves.

A valve consists of a heated cathode, generating free electrons, surrounded by a 'grid', both cathode and grid being surrounded by an anode. The anode is made positive with respect to the cathode, and the negative electrons therefore tend to move from cathode to anode. However, the intervening grid can control this flow if it is made negative with respect to the cathode. If the grid is made sufficiently negative, the flow can be cut off completely. To ensure that the electrons can flow freely, the whole of the valve assembly is enclosed in a sealed evacuated envelope.

A tremendous range of valves with different characteristics were created over the years, but a type which was popular with early computer designers was the 6SN7. This had two valve elements in one envelope, and had a total dissipation of the order of about 6 Watts in a typical application. The binary adder described earlier needed about five such valves.

The transistor looked very different, but could perform a similar service, that of amplifying signals to restore lost power. It was built on a microscopic scale relative to that of the valve, and having no cathode heater it consumed very much less power. The earliest versions were sealed in tiny glass capsules, and were usually soldered directly into the circuit, where valves had usually been plugged into special sockets. Paradoxically, the task of soldering a transistor into its place could cost more than the component itself.

SUSPICION

At first, transistors were viewed with suspicion by engineers who were concerned with equipment reliability. This was perhaps justified by the failure rates shown by early examples, but the doubts persisted. As late as 1959 ther were cries of horror at the suggestion that aircraft engine controls, hitherto based mainly on robust 'magnetic amplifiers', should use the new-fangled devices. Yet the magnetic amplifiers had used diodes built on the same principle as transistors, and the diodes had been accepted without qualms.

Enquiries revealed an intriguing situation. The mortality rate for transistors was said to be high during development, but shrank to a satisfactory level once the engineers had finished poking their measuring probes into the system. The new devices were undoubtedly more fragile than contemporary valves, but would respond to kind treatment.

Reactions to this varied a good deal. The DEUCE, a valve machine, was in full production in 1959, in parallel with fully transistorised machines built elsewhere. The period of overlap between valves and transistors was surprisingly long, and there were a number of reasons for this.

To begin with, the use of transistors called for an entirely new approach to equipment packaging. The traditional method involved mounting the valve-holders on metal chassis formed into an inverted tray. The relatively large size of the valves limited the density of the circuit and the necessary wires could often be connected between the valve-holders. The tiny transistors made this absurd, and an alternative technique was developed on the based of connections etched from copper-clad insultating material.

This settled the method of implementing the circuitry, but new methods of creating the overall mechanical assembly were then

required. This led to many strange schemes before the industry began to reach partial level of standardisation. One key problem was that the circuits themselves could be built at relatively low cost, but the provision of a connector for each circuit unit was expensive, and it also tended to increase the fault rate. Dispensing with the connectors reduced the overall cost, but upset the service engineers, who wanted to be able to remove circuit units quickly and easily.

In short, the advent of transistors caused a major upheavel in electronic design which was not directly connected with the savings in power and heat dissipation which might have been seen as their principal virtues. In 1960, a four-function printing calculator working on a mechanical basis could be purchased for £600, but it was possible to build an electronic equivalent for £1000 ex-works, and that could be linked directly into other electronic equipment, where the mechanical device required complex adaption.

The image of computers as essentially large and very expensive devices was now blurred, with the possibility of relatively cheap free-standing systems which would do a more limited, but nevertheless useful task. Not unnaturally, the established manufacturers decried this approach, and the schism between the 'mainframe' and the small dedicated unit had begun. The consequences of that will be examined in the next instalment.

Meanwhile, the confusion produced by the transistor was increased by the shadow of future developments. As an extension of the search for better packaging methods, small standardised circuit units were produced. These, it was claimed, could be assembled into required configurations in a highly economical manner, cost being kept down by manufacturing the units in large quantities. Unfortunately for the hopes of those who offered such units, there was limited enthusiasm for them, and the total quantities ordered never allowed economic production rates to be sustained.

One reason for the lack of interest in these units was that there were hints of something better to come. A single transistor was so small that several could be fabricated on a single piece of material, together with the interconnections between them. At first, only about fifty components, transistors, diodes, and resistors, were assembled in a single 'integrated circuit', but it was evident that the number would grow in time, though it is doubtful whether anyone then could have envisaged the assembly of ten thousand elements on a single silicon chip.

These integrated circuits were, in concept, similar to the small standardisation circuit which had failed to sell in sufficient quantities, but they had several advantages. They were more compact, more efficient, and they were made in forms of considerable utility. In their ancestry were the standardised gate elements which had been used by Ferranti and by Elliott Brothers, and many similar standards developed on a more localised basis. An odd situation arose, in that those who understood the necessary manufacturing techniques were continually asking circuit engineers what devices they would like to have, but the circuit engineers found difficulty in providing useful answers. They were not yet thinking in terms of integration, and perhaps felt that the new devices might be a threat to their jobs.

The real answer to this was then dimly seen. Whether it was a full-scale computer or a more limited calculator system, the type of device involved was complex enough to need studying at more than one level. Consider, for example, multiplication and division functions. These can be implemented in a number of different ways at the concept level, some offering higher speed, others offering greater detailed accuracy. In those days, one man might find himself responsible for both the concept and its implementation. He needed mathematical and engineering knowledge in combination.

The better solution, eventually adopted in many quarters, was to separate the system and engineering design functions, so that both could be executed by specialists. There was one key difficulty in this, and it was a difficulty that has done the computer industry much harm. If the tasks were to be separated, who was to be put in charge of the whole? This question was complicated by the introduction of the word 'software', in about 1961, which added a third dimension to the matter.

DEFINING ROLES

When a new computing device is to be designed, there are many ways in which the specification can be determined. If the tasks performed are well defined, it may be possible for the 'software engineer' to specify with equal clarity what performance features he needs, but it is more probable that he will nominate a machine which he already knows as a basis for the new design. This is an aspect of the 'Fossilization Syndrome', which has maintained in existance many outdated ideas that would have been better abandoned.

It is equally possible to have a purely engineering definition as a basis, but the best specification is likely to come from the pure systems man, who must mediate between the software and hardware teams. To do this, he must be separated from those teams and given a measure of independence from them. Otherwise, he will be in danger of being accused of bias. As an independant agent, with some understanding of both hardware and software problems, he is in the best position to mediate and guide the overall team.

The transistor continued to be used widely, though not exclusively, at least until the end of the nineteen-sixties. Integrated circuits were appearing, but they were often of local origin and designed for very specific purposes. Some were associated with packaging methods that look incredibly complex in retrospect. They worked, but that was no longer such an important criterion. Effective functioning was beginning to be granted. Economy and efficiency were now coming under closer scrutiny.

One phenomenon of the transistor age was, and still is, viewed with regret. Having based all their early work on prototypes produced by British acadamics, some companies saw fit to base subsequent machines on American designs. This may have been intended to reduce design cost, but if so, the intention failed. Much redesign was needed to produce a satisfactory result, if only because of differing standards and ideas. In one case, at least, a much-improved design was created, almost unofficially, by using an entirely fresh approach.

Looking back on that period, it seems surprising that so much effort achieved so little. The underlying uncertainty bred by the prospect of more and better integrated circuits played a part. It was depressing to put a lot of the effort into a design, knowing that it might well be pushed into obsolescence a year later by some new development. It would have been even more depressing if the full impact of such innovations had been appreciated.

A DECADE'S TECHNOLOGY

Transistors were not the only components which made an impact during the later 'fifties and sixties'. The core store had been mentioned, and this was a major breakthrough. The ability to retain data without power applied was itself a major advantage. The dual nature of the store, usable for permanent or transient data storage, was important in freeing the designer from limitations. The speed of access, no more than a microsecond or so for any data word, was extremely valuable.

Against this was set the fact that core stores were by no means easy to make. For each data bit to be stored, there had to be a tiny ring of ferrite material, the core proper. For a 64K store holding 8-bit bytes, half a million such cores were needed, and these would be arranged in eight 'planes' of 65536 cores each. For each plane, 512 'drive wires' would have to be threaded throughout 512 cores each, and then the 'inhibit' wire would have to pass through all 65535 cores. Bearing in mind that the cores might be no more than a hundredth of an inch in diameter, it can be seen that the store assembly was a task calling for extreme patience and skill. In terms of performance, it remains unique, but in the interests of economy other devices have had to be found to replace it.

This has happened with many of the components and techniques of the computer world. They have come, served their purpose for a while and then have disappeared from the scene. The process still continues, and it would be a bold forecaster who would say with any degree of certainty what marvellous solution will appear next, though it does seem that the process is slowing down in some areas, and perhaps even moving in reverse, as methods which have been tried without success are brought back by those too inexperienced to know better. They will learn, in time, but there is so much to be learned. A lifetime may not suffice.

Lessons of History continues in the next edition of *Computing Today*

ARTICLE SUBMISSIONS

Computing Today welcomes articles submitted by readers, but potential contributors are asked to observe the following guidelines:

- Feature articles should be typewritten with double line-spacing if possible, and should ideally cover between 8 and 16 sides of A4.
- Program listings when absolutely necessary should be kept short. When listings are supplied they must be accompanied by text describing the algorithms employed and, if possible, a flowchart.

Articles should be sent, along with an accompanying letter describing the nature of the article, to:

SUBMISSIONS
COMPUTING TODAY
1 Golden Square
London W1R 3AB.

Please ensure that your name and address is printed clearly on your covering letter.

READERS

HELP US
TO HELP YOU:

MENTION

COMPUTING TODAY

WHEN RESPONDING
TO ADVERTISEMENTS

AMERICAN & BRITISH
COMPUTER
BOOKS & MAGAZINES

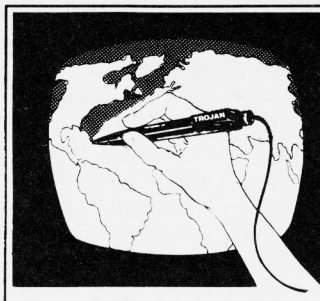
We offer a World-Wide service for books from all publishers and more than 50 magazines.
Please fill in the coupon and ask for monthly listing of NEW books and magazines. You may also subscribe to the following popular magazines starting from the current issues.

	Europe	Other
AMSTRAD CPC 464 USER	£19	£27
BYTE	£26	£32
COMMODORE USER	£18	£28
MSX COMPUTING	£18	£28
MSX USER	£18	£28
PERSONAL COMPUTER WORLD	£19	£28
QL USER	£18	£28
SINCLAIR PROGRAMS	£18	£28
SINCLAIR USER	£19	£29
YOUR COMPUTER	£21	£32

These are SPECIAL 6 months Trial Offer Prices.
Please fill the coupon and send it with your cheque to:

COMPUTER BOOKS AND
MAGAZINES — HAULSTERM LTD
Dept CT7 Exeter House, Putney Heath, London SW15
Tel: 01-788 8746

Name ..... Tel: .....
Address .....
.....
I wish to subscribe for 6 months to the following magazines:
1 ..... £ .....
2 ..... £ .....
3 ..... £ .....
Enclosed please find a cheque for Total £ .....
Please send me list of books and other magazines.



LIGHT PENS

COMMODORE 64
VIC 20 (NEED SUPER EXPANDER)
SPECTRUM 48K
DRAGON 32/64
TANDY COLOUR 32K

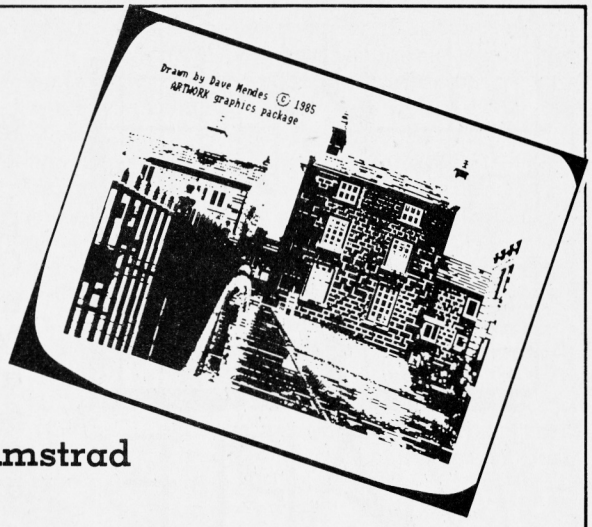
NORMAL PRICE
£17.25
LIMITED PERIOD OFFER
£14.95
inclusive
NO INTERFACE REQUIRED

Discover the exciting world of creating your own graphics on screen.
The Trojan Light Pen will draw boxes, circles, lines, freehand pictures, save and load pictures with full erase facility.
All in Hi-Res screen in any of 4 colours for the Dragon/Tandy, 8 colours for the Spectrum and Vic 20, and 16 colours for the Commodore 64.
For educational or leisure use.

TROJAN
Micro Computer Software & Accessories

Send cheque/P.O. to.
TROJAN PRODUCTS
166, Derlwyn, Dunvant, Swansea SA2 7PF
Tel: (0792) 205491.
TRADE ENQUIRIES WELCOMED

KUMA'S ARTWORK



Bill Horne **reviews a graphics utility for the Amstrad CPC464**

Drawing pictures on the screen of an Amstrad CPC464 is a matter of setting some or all of the 16,000 screen locations (384 are not used at a given time) and this can be quite a chore, especially in the abscence of CIRCLE and FILL commands. As a solution to the problem, KUMA offer 'ARTWORK', a graphics utility package.

An interesting feature of the program is that you can check through a series of menus without losing the picture you are working on. This must surely mean that two screen RAM areas are used, one at 4000 - 7FFF and the other at C000 - FFFF, an extravagance that would make users of some other computers green with envy, but which leaves plenty of spare room on the Amstrad.

Press COPY, and the available palette colours are displayed across the top of the screen. You can pick a colour by using the left and right arrow keys, or change it with the up and down arrow keys. Press COPY again, and you're in drawing mode.

A line can be drawn by holding down SHIFT and using the arrow keys to move the cursor. Without Shift, you can move the cursor without drawing. A more interesting option, however, is obtained by using the arrow keys with CTRL pressed. This draws a trial line, which you can erase by pressing SPACE. Otherwise, the line becomes permanent when you release CTRL. The plot colour is always shown in the Border.

Press F, and the area containing the cursor is filled with the current colour. If there are any gaps in the containing boundary, and they are not always easy to avoid, the fill may 'leak', coving a wider area than you intend, but pressing U will remove the fill (Unfill) and allow you to mend the leak and try again.

Press B, and a box outline appears. You can change the dimensions of the box or move it about, but nothing permanent will be drawn until you press Enter, when the box is drawn and filled.

Press C or E, and a circle or ellipse appears. You can change the dimensions or move the shape, but the permanent drawing does not appear until you press Enter. You can then fill the shape if you wish.

Press T, and you can insert text, with DEL effective to erase errors. The arrow keys will move the completed text to whatever position you want, and Enter will make it permanent. (Though you can always erase anything by drawing over it in background colour.)

FACTSHEET:	Artwork
Price:	£9.95
Available from:	Kuma Computers Ltd. 12 Horseshoe Park, Pangbourne RG8 7JW Tel: 07357 4335

With S pressed, the arrow keys will scroll the whole picture up, down, left or right, with full wrap-round in all directions.

The remaining facilities are covered by the menus, and there is a 'help' page for each main function. You can load a picture from tape, or save it to tape, and the program is claimed to be compatible with the disc drive system. Saving speed can be set fast or slow, and CAT is available. Screen mode can be changed.

The tape record consists of two files, one defining colours and other key parameters, the other holding the picture. The pictures can be incorporated in other programs at will. If the colours and parameters are set by the program, the first file is not needed.

We had few complaints, though it seemed that some key funtions were reluctant to respond, perhaps because of long polling time, whereas the arrow keys were sometimes rather prompt, a very quick dab being needed to make a single step change.

A very impressive demonstration picture is provided, using Mode 2, and looking like a black and white engraving. If you want that kind of detail, you have to accept two colours, whereas the sixteen-colour mode can be a little crude in its outlines. You have to make a compromise.

If one were really to carp, it might be said that there is occasionally evidence that alignment with the groups of bytes used in controlling colour is sometimes lost, the evidence being a narrow vertical band of odd-ball colour, but that is rarely noticeable or a problem.

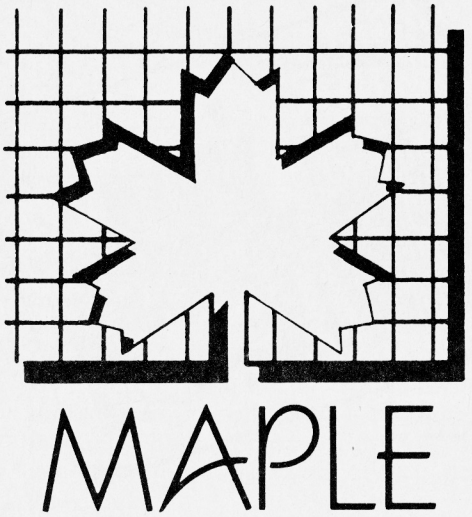
A nice program that will keep you happy or frustrated for hours.

```
Circle drawing

Selected     by using <C> key
Move circle  using cursor keys
Expand       using <SHIFT> + ↑
Contract     using <SHIFT> + ↓
Draw circle  with <ENTER> key
Abandon      by pressing <SPACE>

Press <ESC> to return
```

Circle drawing 'help' page: each Artwork option has an associated help page



Graham Marshall, G F Marshall Computer Services.

Both the Forestry Commission and the Nature Conservancy Council are using digital maps to detect changes in the English landscape. Mapping from Aerial Photography and Land area Evaluation, or MAPLE, is a sophisticated, computer-based mapping technique which may eventually be accessible to all.

And again the Devil took Him up into an high mountain and showed Him all the Kingdoms of the world and all the glory thereof.'

Ever since I was a child I have always been fascinated by this verse from St. Matthew. How high was the mountain? How did the Devil deal with the curvature of the earth? How do you show off glory? It is only recently that I have realised how it was done. The Devil had a PC on top of his mountain and showed off the world using the facilities now available via digital mapping techniques.

I am sure readers are familiar with the Ordance Survey and the National Grid. The National Grid superimposes a 3 - dimensional cartesian coordinate system, based on a false origin to the South West of the Scilly Isles over the whole of Great Britain. Any point can be identified by giving a grid reference of eastings and northings and, if necessary, a height above sea level. Most school children learn the techniques of extracting a grid reference. The familiar O.S. Landranger series of maps, formerly the 1 inch to 1 mile, but now drawn to scale of 1:50000, enable the user to identify a map feature to the nearest 100 metres. At larger scales it is possible to identify a feature to the nearest metre, or even down to a millimetre if required.

Every country in the world has its own national grid, or shares one with its neighbours, or could have one imposed on it. By holding sets of point references (eastings, northings and height) together with a classification code it is possible to define the line of a major road, or a garden fence, or to pinpoint a particular manhole cover. Store this information within a computer and we have the basis of digital mapping.

Mainframe and mini systems have been around for many years and the commercial mapping companies all use them. Those who were early in the field had bespoke systems developed and more recently packaged systems have become available. The most sophisticated systems use multiple graphics screens to show maps in colour or at different scales and to generate terrain models (the table cloth over the hillside picture which produces a 3 dimensional representation of the landscape). These machines are coupled to large plotters, either digitising tables or encoded stereo photogrammetic plotting instruments to input map information, and to equally large output X-Y plotters to produce maps.

Automated capture methods are now being developed, and the Ordance Survey hope to complete the digitisation of all the 300,000 maps and plans they produce by the year 2000. The digitised information for those parts of the country already covered is available on magnetic tape from the Ordance Survey and is used by public utilities, local authorities and government bodies to output maps and for other purposes.

THE MAKINGS OF MAPLE

My company, G F Marshall Computer Services, won a contract from the Forestry Commission of Great Britain in 1982 to develop a digital mapping system for them using micro computers, and all of the illustrations of this article have been generated on this system, or a new version developed for the Nature Conservancy Council.

The Forestry Commission system was dubbed MAPLE (Mapping from Aerial Photography and Land area Evaluation). The MAPLE system allows forest map information to be extracted from pairs of aerial photographs using encoded stereo plotters. These machines generate an accurate 3 dimensional image of the ground that an aircraft has flown over and photographed. This 3 dimensional image is known as a model, and the plotter operator may locate points and follow linear features accurately in 'model space'. The locations of these points and lines are captured by encoders fitted to the stereo plotter which are monitored by a device known as SID (Stereo Interface Device). The SID passes the encoder readings to the micro computer where they are transformed into grid coordinates.

The grid coordinates are stored on floppy diskette and a spatial index entry created for each line and point feature. A spatial index allows the (map) information for a particular area of land to be extracted more rapidly than a serial review of all the data: one doesn't need to consider Cardiff or Birmingham if you are interested only in London and the South East. As the grid coordinates are stored, a map is displayed on the VDU screen of the micro. The particular micro chosen for the projects have been those with the best screen resolution available at the time. Indeed, the machine used for the Forestry Commission project is still ahead of many later and more commercially sucessful micros - such is the waywardness of our business.

The screen map window may be changed and the scale of the map displayed altered to highlight the area where data is currently being captured from a series of plotter models and will resolve the edge matching automatically. Similarly, data may be captured from existing maps via a digitising table and linked with data from the aerial photography. Eventually all the data for a particular forest management area will have been captured and a map will be output on an X-Y plotter.

OTHER FACILITIES

If this was all that MAPLE did, then we would have very little to write about. But the system provides several more facilities.

Firstly, the Map information may be edited. This is a two stage process. Theoretically the plotter operator should declare a 'node' where two or more lines cross, and should plot with sufficient accuracy to ensure that T junctions, for example, close-up exactly and not overlap or fall short. This is very important when polygon identification takes place since the computer will very quickly find 'holes' in the line work and create larger polygons than were intended. MAPLE therefore takes the map data and the spatial index and automatically reviews it all. Where nodes should have been declared MAPLE will create them.

Where two or more nodes may be superimposed, MAPLE will combine them. Where a potential T junction, or an overlapping T junction occurs, MAPLE will bend the top of the T to exactly meet up with the down stroke.

The program will also identify any of 'potential errors' in the data. Such errors include duplicating lines and loose ends. Details of these potential errors are then presented to an operator in the Manual Edit program. This program draws a map on the micro VDU screen and allows the operator to amend the data by adding, deleting or extending lines, by changing line designation, and by editing the single point data. The operator corrects the potential errors found by the automatic review or declares them to be valid, he may also amend the data where necessary in the light of his own observations of the map data.

The edit process is concluded by the addition of any map narrative information required. A map is of no use whatsoever if it comprises only line work; one wants to know the name of the street, the height of the hill or where Mr Brownlow lives. These map narratives are entered via the keyboard and are positioned on a map drawn on the screen. The latest version will allow a narrative string to be searched for and will then draw a map on the screen centered on the narrative. The hours spent searching the index of the A-Z to find where Accacia Avenue is now a thing of the past - although we still need someone to come up with a micro with a decent graphics screen to slip into your jacket pocket.

POLYGON IDENTIFICATION

Back in the forest, MAPLE automatically identifies polygons formed by particular sets of lines. This process involves selecting a start node and a line leaving that node, then 'turning right' at each subsequent node until one arrives back at the start node. Each internal line has to be traversed twice, once for the area on the left, once for the area on the right. Readers may like to draw a small matrix of lines to try out the technique for themselves. This is a technique which is easily enough to describe in words and simple to demonstrate visually on paper, but it is not so simple when it comes to programming. How do you select the 'next on the right' when you have 5 or 6 lines radiating from a node?

Having identified the lines which form the boundary of a polygon, MAPLE then calculates its area and selects a point with the polygon boundary. The area may have to be amended subsequently in case a particular polygon contains further 'island' polygons.

Once polygon identification is complete the operator is then invited to link the polygons together to form forest management compartments. These compartments are typically 10-20 hectares (1 Hectare = $10m^2$) in extent and may contain several different varieties of trees. The polygons are formed using natural boundaries such as internal forest roads, or the boundary between one stand of trees from another (oak from ash for example) or by the artificial boundary imposed by legal lines distinguishing areas owned by the Forestry Commission from those of other landowners.

As the compartment linkage takes place the operator may add map narrative, which as well as having a map location, is associated with the individual compartment.

All this information is output in the form of a final forest map which is distributed to the forester on the ground (and is available from the Forestry Commission for recreational purposes.) The forester uses the map to determine which stands of trees should be felled, which thinned, and where new planting should take place; and every year the map has to be updated to reflect the changes on the ground.

The techniques we had developed for the Forestry Commission MAPLE system came to the attention of the Nature Conservancy Counsil. They, along with various other Governmental bodies, had been charged with the task of measuring land - use change. How many hectares of dappled broad - leaf English woodland have been replanted with plantations of dark, inpenetrable, conifers? How many miles of hedgerow, the home of song birds, wild flowers and the very essence of England, have been grabbed-up to create featureless prairie farms? How much wild, open, empty fen and marsh has been drained to leave mundane sugar beet fields? These, and many similar questions have been asked of successive Ministers of Agriculture, who have been unable to provide accurate answers. The questions are usually phrased in emotive terms (like my sample ones here) and asked by professional prophets of doom. How do you go about providing an answer?

There exists an archive of aerial photographs of Great Britain dating back to 1945 There are even earlier ones taken by the German Luftwaffe before and during the war. By using the photos taken in 1945/46, and others taken in 1980/81, and the techniques we have developed for the MAPLE system, the same sample areas can be plotted for the different time frames as follows:

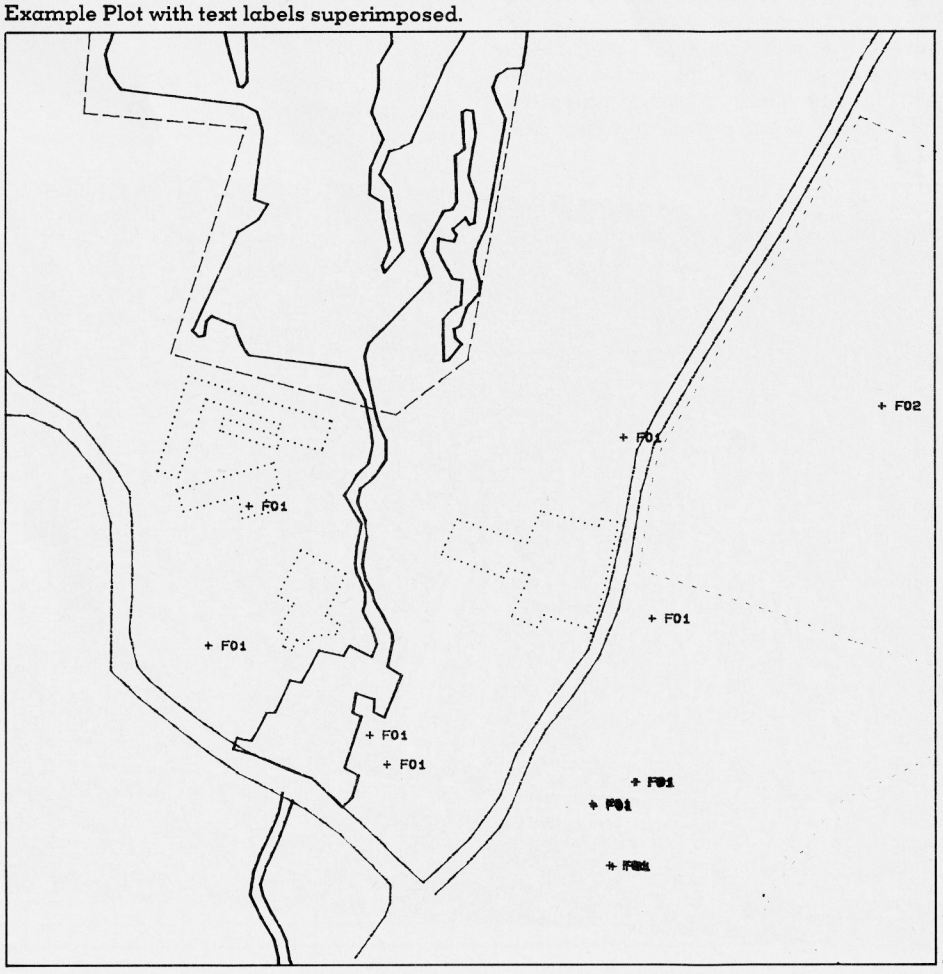
- land use is assigned at the capture stage by identifying a land use classification at a point within each land parcel.

- data is transformed, stored, indexed, edited, plotted and 'polygonned' in a similar manner to the MAPLE system; although the computing equipment used is now a 16 bit micro running under Concurrent DOS compared to an 8 bit machine running CP/M.

- New software now takes each land use classification point and assigns it to the polygon in which it is found. An error reporting technique similar to that detailed above for the line work highlights and obtains resolution for those polygons which have not been given a land use classification at all, and also those which have several conflicting classifications.

- By using a pixel technique, the land use from the two time periods is determined and a matrix of results showing old land to new land use in area terms is established. The comparison program allows the pixel size to be varied from run

Example Plot with text labels superimposed.



Example Plot showing land divisions.

to run. Being an area based activity halving the pixel size quadruples the number of comparisons to be done.

A similar technique is used to determine changes in linear features. It is not sufficient to report that since 1945, 15 miles of hedgerow have been reduced to 8 in 1985. How many miles of the 8 were original, and how many are new? A hedgerow, being a living entity, may 'move' in 40 years. This, coupled with the errors introduced due to the age and condition of the earlier photography, means that an old hedgerow will not necessarily coincide linearly with its thriving modern self in purely algebraic terms.

RESULTS

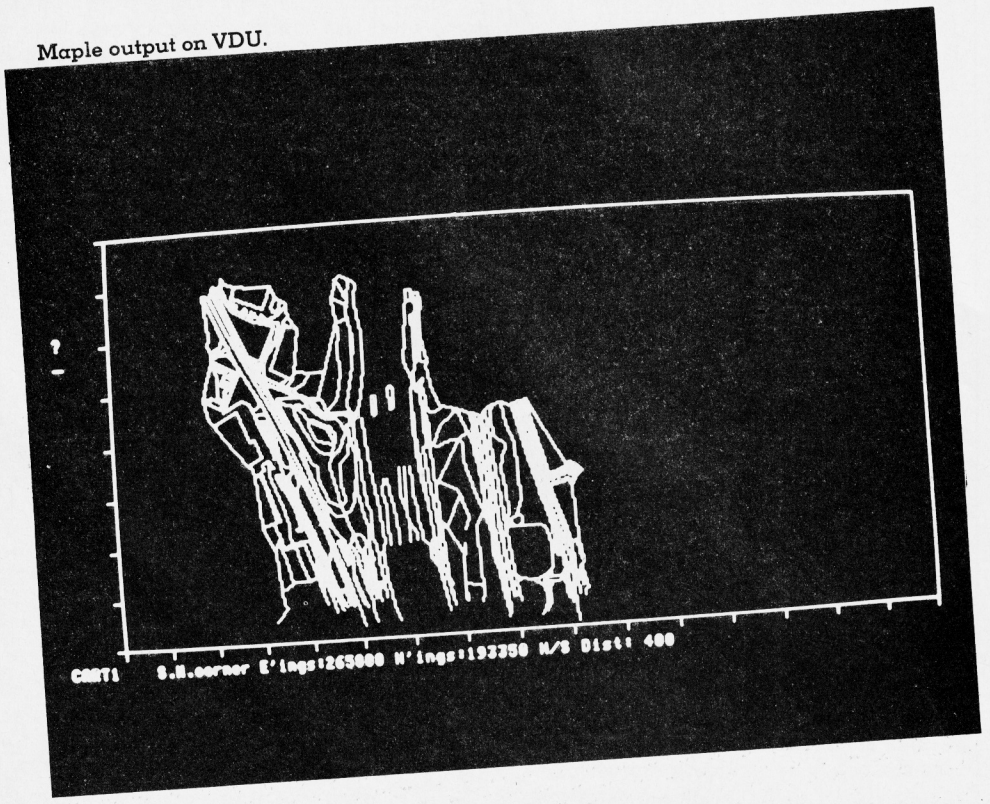
The full results of all the measurements taken by the Nature Conservancy Council are to be published over the next 10 years on a county-by-county basis, and monitoring of the countryside is to continue well into the next century.

The sophistication of the software we, and other companies, have developed means that town maps may now be produced showing the area of every individual house plot, the names of the house holder and other occupants, and the rateable value. Such maps are of great use to town planners, police forces and social services. Again, Service Mapping, where the position of the telephone, water, sewage, gas and electricity services in the road is shown in detail, will eventually save many millions of pounds per year. By providing up to date information as to where each service is to be found, the gas men will no longer dig up the telephone wires, nor the electricity men the sewers. Since these maps may be output on a VDU screen; or quickly printed out, used to locate the problem, and then thrown away; the demand for up-to-date map data in digital form will continue to grow.

We have shown that Digital Maps on micros are a practical proposition, and we hope that in the next 5 years every serious map user will have access to a micro and mapping software, and that digital maps will be commonplace. Already our digital forestry maps can be used to generate terrain models. More sophisticated software would show the types of trees positioned on the terrain model and 'growing'. The forester can then determine his felling and thinning operations to minimise conflict with the conservationist.

Digital maps can be used to highlight and illustrate information that is geographically based. Maps to show the level of unemployment, the distribution of nuclear warheads or the proportion of undernourished children in a country can all serve to illustrate the current state of the Kingdoms of the World and the glory thereof. Assuming that Christ saw those maps he was very right to turn the Devil down.

Maple output on VDU.



COMPUTING Today

NEXT ISSUE OUT FRIDAY JULY 12TH

UNIX — We begin a four part series describing the intricacies of Bell Labs' multi-user programming environment.

ADA — First Publishing's ADA for the Commodore 64 gets the once-over by our reviewer.

DOMESDAY — Computing Today takes a look at the BBC's project to revive William the Conquerer's inventory.

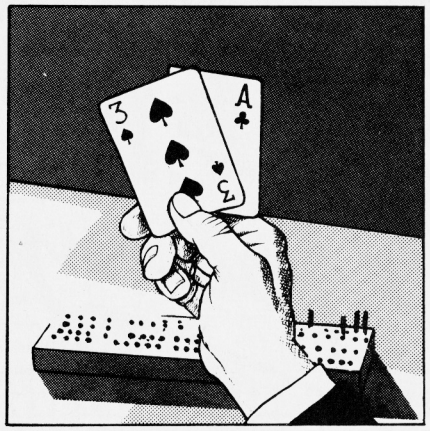
SHORT STORY — The Case of the Moriarty Engine — Holmes and Watson become suspicious when the evil Moriarty's fortune on the stock market takes a sharp turn for the better.

All this and more in the next edition of

Computing Today

Articles described here are in an advanced state of preparation but the circumstances may dictate changes to the final contents.

THE EVOLUTION CONTINUES..



Original program: Bob Stafford, Australia
Adaption: Computing Today Staff

We begin here a series of articles discussing the decisions involved in writing a BASIC program. The fact that the program plays the game of cribbage is relatively unimportant. The key point is that a number of fairly complex processes have to be performed, and the necessary routines have to be created and linked together. Versions are derived for the BBC computer and the Amstrad CPC464, providing an interesting comparison between these two machines.

CRIBBAGE PLAYER

The subject of the program for this series more or less chose itself. We wanted something reasonably complex, and in leafing through our collection of listings we came across a version of Cribbage Player written by an Australian called Bob Stafford back in 1980. It seemed exactly what we wanted, so we got in-touch with him - not without some difficulty - and he agreed to let us use the relevant parts of his routines. For that, we are duly grateful, because the program has always interested us, even after five years of intermittent use.

Cribbage is reputed to be the oldest card game still in popular favour It is said to have been invented be Sir John Suckling in the early part of the seventeenth century, and the odd phrase 'One for his Nob' may be an abbreviation for 'One for his Nobility', a reference to the inventor. Although superficially simple, the game has a complex scoring system which needs careful concentration.

RULES OF CRIBBAGE

The game begins with a cut from deal, the low card winning. Six cards are then dealt to each player, and each player discards two of these to form a 'crib'. After the discards have been made, a thirteenth card is exposed as the 'starter' or 'turn up'. Should this be a Jack, the dealer scores a point.

The 'lay out' phase of the game then begins. The players, non-dealer first, lay cards face up on the table alternately. If the total value of the cards laid is fifteen, the last to lay a card scores two. The total value of the cards on the table must not exceed thirty-one.

If a player has no card which will keep the total within that limit, he must call 'GO'. When neither player can continue, the last to lay a card scores one point. If there are any cards left, the layout procedure begins again.

Incidentally, while the cards from ace to nine have their normal values, Ten, Jack, Queen and King all have a value of ten in this context, while retaining their individual denominations for other purposes.

Other scores in this phase will be examined later.

When all the cards have been laid, the 'show' phase begins. The three hands are exposed and scored — first the non-dealer's, then the dealer's, and lastly the 'crib' hand, which benefits the dealer. In this phase, the turnup card is considered to be added to each hand in turn.

The process is then repeated, the deal changing hands. When either player reaches the agreed winning score, which maybe 60 to 120, the game is over.

A particular feature of the game is the 'Cribbage Board', shown in Fig 1. Four rows of thirty holes each are drilled in a slat of wood, and the score is kept by moving small pegs along the rows. Two pegs are used for each player, and these leap-frog past each other, so that both the present and last scores are shown. This reduces the chances of a little gentle fiddling...

The pegs move from left to right on the outer rows, and from right to left on the inner rows.

THE DISPLAY

We now have enough material to provide a bases for designing the display. There is quite a lot to be fitted in.

The cribbage board has thirty-five holes or spaces in its length, and with a forty-column display this is just convenient, but simulation of the board on a thirty-two column display would present greater problems. It would be possible, but highly inconvenient, because graphics coordinates would have to be used. The original program was for a sixty-four column screen, by the way. That presented no problems at all.

Apart from the scoreboard, we have to find room for twenty-one card positions; six for each hand as dealt, eight for the table layout (though that maximum is not often reached, it does arise now and then) and one for the turnup card. Storing column and row positions as separate items is one approach, but array space and data space can be saved by storing 40*Line + Column. This means that the two values have to be worked out when the data is used, and there are arguments in favour of either approach. We have used the single variable method here, but you may prefer to change that.

Finally, a message area must be reserved for messages and prompts.

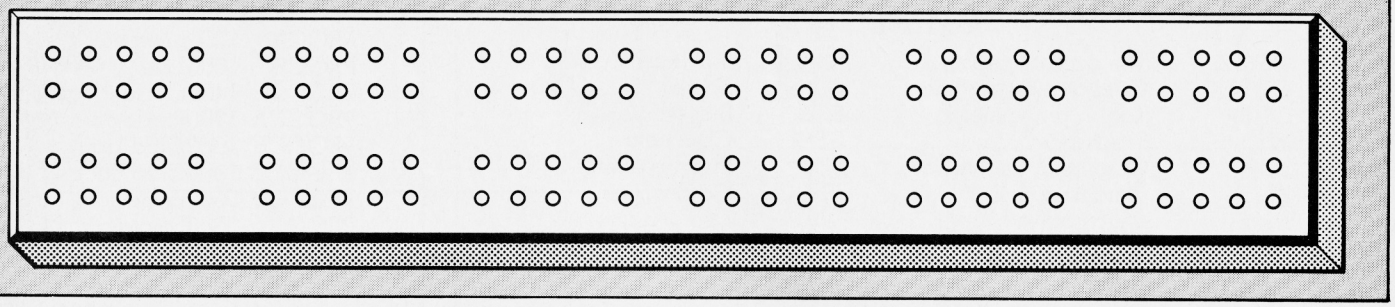
It has been assumed that the display will mainly be built up from character shapes, since drawing nice card outlines in graphics can become rather complicated and tedious.

A possible arrangement is shown in Fig 2. There is not a lot of room for manoeuvre. To ensure against scrolling accidents, the CPC464 version needs to establish a window on the bottom line, forming the message area. The BBC version has a little more room vertically.

At this point, however, we have a little heart-searching to do. It would obviously be nice to show the red and black suits in their correct colours, but that would call for three or four colours. The CPC464 can provide that easily enough, but not the BBC Micro. The program is going to be quite sizeable, and Mode 1 only leaves about 8700 bytes of program area. Mode 7 will not provide the character patterns needed, and Mode 6 will not give vertically-continuous character shapes, so we are left with Mode 4, and two colours.

A similar problem arose with the 'Poker' program published last year. The awkward point is that it is difficult to be sure, before the

Figure 1: the Cribbage Board



program is written and entered, how much store space will be needed. On the whole, it is better to be pessimistic, rather than optimistic, since that is less likely to lead to frustration later...

Before leaving the subject of display, it should be said that card positions 15 - 19 can be used in the 'show' phase, and positions 11 and 18 will serve to display the cards in cutting for deal. The latter pair are not quite central, and purists may care to set up central positions for the purpose.

TITLE DISPLAY

The original program had an ingenious and attractive title display, with the letters enclosed in card outlines, and an outer frame of suit symbols. More to the point, it invited the input of a number to be used as a basis for the random number system. Without that, every game would at least have started in the same way. We have made randomisation automatic.

It is usually convenient to execute the initialisation routine while the title is being displayed. The saving in time may be minimal, but it is always useful to get the computer doing more than one thing at a time.

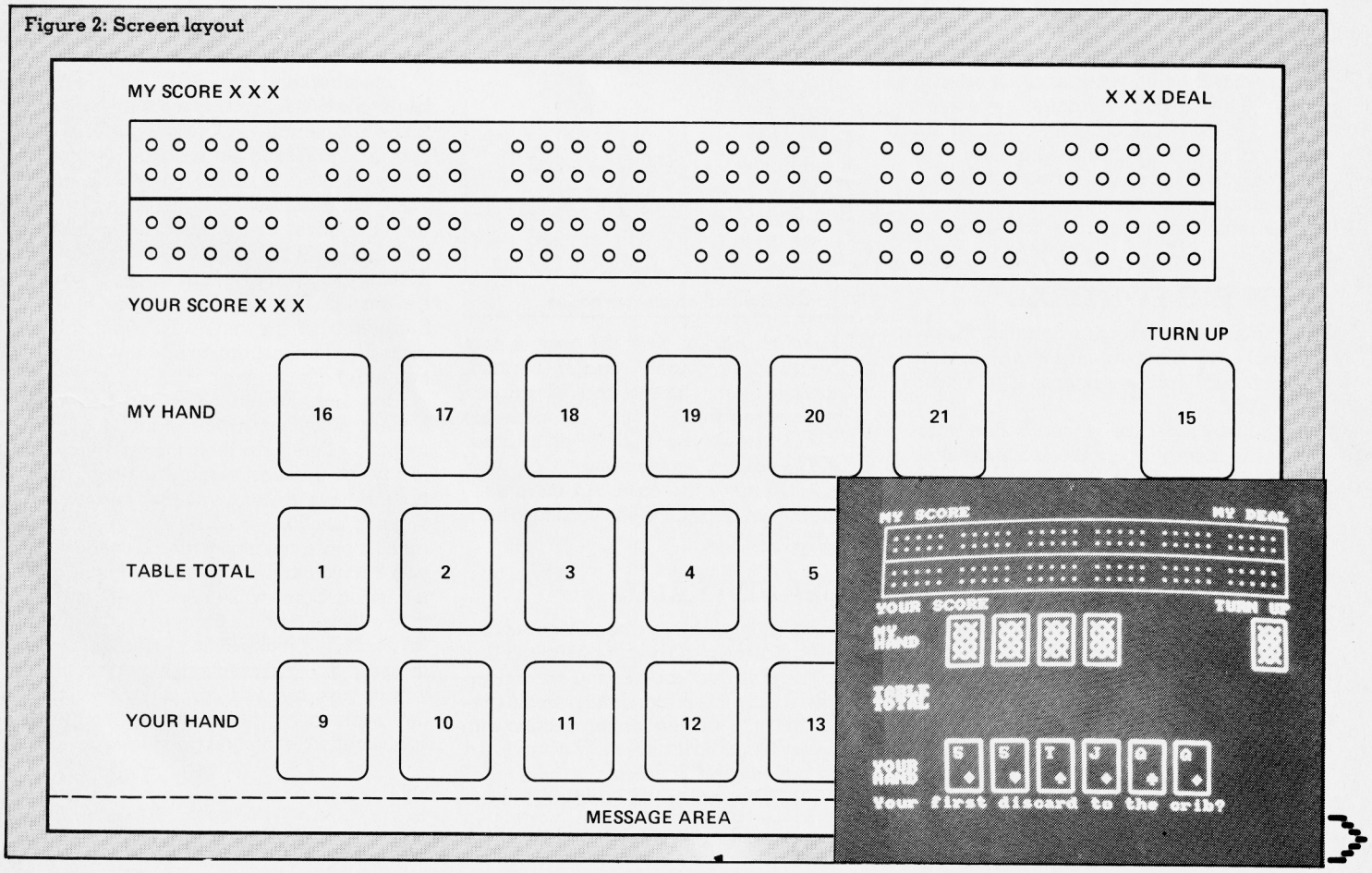
The simple title routines are shown in Listings 1a and 1b for the two machines. The BBC version uses Mode 7 and double-height characters, the CPC464 version is more restrained but quite effective. In both cases, a delay routine is provided. It will be used by other parts of the program in due course.

INITIALISATION

Before the program is finished, we will need a number of arrays, some of which cannot be justified at this stage. For the BBC version, all arrays have to be dimensioned, but the CPC464 does not require this for arrays with a subscript range of ten or less. We have left REM lines to allow additional dimensioning to be added later, but the arrays shown are for use as follows:

- We start with a pack of 52 cards, and we will need to know which cards are left in the pack at a given moment. CC(52) serves this purpose.
- The cards will initially be defined by single numbers, which will be split into denomination and suit. However, as noted earlier, the 'value' of a particular denomination is not always the same as the corresponding name, and that calls for a pair of arrays, N$(13) to give the single-letter name, and VN(13) to give the value.
- Similarly, the suit numbers will need to be translated into names, again in single-letter form, and also into suit symbols. For this, we have S$(4) for the name, and SG$(4) for the symbol.
- CP(21) will store the twenty-one card positions.

Figure 2: Screen layout



Arrays

CC(52)	The pack of cards	S$(4)	Suit letters
CP(21)	Card display positions	SC(2)	Scores
CT(8)	Table card values	SG$(4)	Suit symbols
HH(3,6,2)	The hands	ST(9)	Card hold
HN(5)	Card hold, denomination	TU(2)	Turn up card
HS(5)	Card hold, suit	VN(13)	Card values
N$(13)	Card names	W$(2)	Dealer name
PC$(6)	Card strings	ZH$(8)	Scoring reasons

Variables

A$,B$,D$ etc	String holds	M1$, M2$	String holds
AA, BB	Card data holds	N	Character position
B1, B2	Back peg score	PF	Show hand count
C	index	PL	Player
C1	Cards in player's hand	PT	Table total
C2	Cards in computer's hand	Q$	String hold
C8	Cards on table	QP	Index
DD	Cards in pack + 1	SC	Score
DL	Dealer	SK	Hand pointer
EF	Validity flag	X	Column
F1, F2	Front peg scores	Y	Line
F3	Score hold	ZH	Score text pointer
GF	GO flag	ZJ	Jack score
HP	Show hand pointer	ZF	Flush score
HT	Score hold	ZP	Pair Score
J5	Number of cards to sort	ZR	Run score
K	Card pointer	ZZ	Run score hold
LS	Largest score		
M	Delay parameter		

PROCEDURES

PROCA	1200Initialise
PROCB	1000Title
PROCC	1400Cut for deal
PROCCA	1700Pick a card
PROCCB	1800Card Frame
PROCCC	1900Card Back
PROCCD	2000Erase card
PROCD	2300Display Frame
PROCDEL	1100Delay
PROCE	2430Lower Frame

- HN(5) and HS(5) will initially be used to store the name and suit of the cards cut, but there are a number of instances where it is convenient to store sets of five cards separately.

The special characters required are listed in Fig 3. All the special characters have to be created for the BBC computer, but all are standard for the CPC464. (That said, you may feel you could produce better shapes in some instances by brewing you own...)

The initialisation routines are shown in Listing 2a and 2b.

CUTTING FOR DEAL

For what it does, the routine handling the cut for deal may seem rather lengthy, but its subroutines are extensively used elsewhere.

The first procedure or subroutine introduces the interesting card selection routine (PROCCA and 1700). By comparison with some routines for this purpose, Bob Stafford's scheme is neat and elegant.

Initially, CC(52) holds the numbers 1 to 52. A change to the range 0 - 51 would simplify matters in some respects, but would complicate other aspects. DD is initially 53. Whenever PROCCA is called, DD is decremented, and a random number between 1 and DD (inclusive) is created, this being F%. CC(F%) is read, and CC(F%) = CC(DD). When the next card is drawn, F%. will have the range 1 to (DD - 1), so CC(DD) would be ingnored. By using it to overwrite the card which was taken, the full range of cards remains available, as it should be.

Shape	BBC	CPC464
Scoreboard Hole	224	144
Scoreboard Peg	225	231
Top left corner	226	150
Top right corner	227	156
Bottom left corner	228	147
Bottom right corner	229	153
Top/bottom	230	154
Sides	231	149
Card Back	232	203
Hearts	233	228
Spades	234	229
Clubs	235	226
Diamonds	236	227

Figure 3: Special character codes

The number taken from the array is split into NN (the denomination) and SS (the suit), allowing transfer to the HN and HS arrays.

There are three other procedures: PROCCB draws the outline of a card, PROCCC draws in the card back, and PROCCD erases the card. An extra subroutine is used in the CPC464 version to handle card colour.

DISPLAY FRAME

To complete this stage, we need to put up the framework of the main display. This falls into two parts: The scoreboard and the turnup card (which will be added later) remain present all the time, but the section dealing with table play is erased for the 'show' phase, and has to be renewed thereafter.

PROCCD handles the fixed part of the display (see Listing 4), and calls PROCE to deal with the part that has to be renewed. There is nothing very special about these routines.

SUMMARY

We now have an initial group of subroutines which will execute the cut for deal, the title display and the main display framework. The listings for the two machines show some interesting differences, on which a few comments are needed.

The fact that the CPC464 routine has 73 lines and the BBC version has 96 lines is not too significant, since it arises mainly from the happy fact that the CPC464 can provide the special characters required, so that there is no need to set them up.

The advantages of the procedure-definition system of the BBC machine are not outstanding here. It is true that a procedure name can be written-in without wondering where the definition will go, whereas a GOSUB instruction cannot be completed until the position of the subroutine has been determined, but the inconvenience of that is minimal.

Purists contend that GOTO should never be used in BASIC, but it is sometimes difficult to see an alternative. Consider the GOTO at line 1780 in Listing 3a. A check is made for card equality at line 1750. We want to repeat the section consisting of lines 1630 -1740 until the cards are unequal, but REPEAT UNTIL cannot be used, because there is a message to go out first, and the cards have to be erased. The GOTO solution is neat. Supporters of the anti-GOTO school are invited to demonstrate an equally neat solution using other means.

One thing should be added before we close-down this first part of the study. The routines offered do their job, but may not always be the most elegant solutions available. Here and there, you may see room for improvement. It would be surprising if you didn't. Before you malign us, however, make sure that your alternative does exactly what is needed. Some while back, a reader wrote in to criticise a prime number routine which we published, quite failing to realise that his solution did not create the list of primes which was needed for a further stage. Improvement by changing the performance specification is rarely a good idea...

Part 2 of Cribbage Player appears in the next edition of **Computing Today**

```
1000 DEFPROCB
1010 MODE 7
1020 PRINT TAB(9,12);CHR$(134)CHR$(141);"Cribbage Player"
1030 PRINT TAB(9,13);CHR$(134)CHR$(141);"Cribbage Player"
1040 PROCA
1050 PROCDEL(300)
1060 MODE 4
1070 ENDPROC
1100 DEFPROCDEL(M)
1110 FOR X=1 TO 20*M:NEXT
1120 ENDPROC
```

Listing 1a: title

```
1200DEFPROCA
1210DIM HN(5),HS(5),VN(13),N$(13),W$(2)
1220DIM CC(52),CP(21),S$(4),SG$(4)
1230REM
1240REM
1250VDU 23,224,0,0,0,&18,&18,0,0,0
1260VDU 23,225,0,0,&3C,&3C,&3C,&3C,0,0
1270VDU 23,226,0,0,0,0,1,6,8,8
1280VDU 23,227,8,8,8,6,1,0,0,0
1290VDU 23,228,0,0,0,0,&C0,&30,8,8
1300VDU 23,229,8,8,8,&30,&C0,0,0,0
1310VDU 23,230,0,0,0,0,&FF,0,0,0
1320VDU 23,231,8,8,8,8,8,8,8,8
1330VDU 23,232,&81,&42,&24,&18,&18,&24,&42,&81
1340VDU 23,233,0,&66,&FF,&FF,&7E,&3C,&18,0
1350VDU 23,234,&10,&38,&7C,&FE,&FE,&7C,&10,&3B
1360VDU 23,235,&1C,&1C,&6B,&7F,&6B,8,8,&3C
1370VDU 23,236,&18,&3C,&7E,&FF,&7E,&3C,&18,0
1380FOR J=1 TO 13:READ N$(J),VN(J):NEXT
1390DATA A,1,2,2,3,3,4,4,5,5,6,6,7,7
1400DATA 8,8,9,9,T,10,J,10,Q,10,K,10
1410FOR J=1 TO 4:READ S$(J),SG$(J):NEXT
1420DATA S,CHR$(234),C,CHR$(235),D,CHR$(236),H,CHR$(233)
1430FOR J=1 TO 21:READ CP(J):NEXT
1440DATA 608,612,616,620,624,628,632
1450DATA 636,808,812,816,820,824,828
1460DATA 435,408,412,416,420,424,428
1470REM
1480REM
1490REM
1500X=TIME :P=RND(-X)
1510VDU 23,1,0;0;0;0
1515 W$(1)="You":W$(2)="I"
1520ENDPROC
```

Listing 2a: initialise

```
1600DEFPROCC
1610PRINTTAB(5,3);"C U T T I N G   F O R   D E A L"
1620DD=53:FOR A=1 TO 52:CC(A)=A:NEXT
1630FOR J=1 TO 2:PROCCA:HN(J)=NN:HS(J)=SS:NEXT
1640PROCCB(18):PROCCC(18):PRINT TAB (10,10);"My card"
1650PROCCB(11):PROCCC(11):PRINT TAB (10,20);"Your card"
1660PROCDEL(100)
1670PROCCD(18):PROCCB(18)
1680PRINT TAB(18,13);SG$(HS(1))
1690PRINT TAB(17,11);N$(HS(1))
1700PROCDEL(100)
1710PROCCD(11):PROCCB(11)
1720PRINT TAB(18,23);SG$(HS(2))
1730PRINT TAB(17,21);N$(HS(2))
1740PROCDEL(200)
1750IF HN(1)<>HN(2) THEN 1790
1760PRINT TAB(12,5);"We'll cut again."
1770PROCDEL(300):PROCCD(18):PROCCD(11)
1780GOTO 1630
1790IF HN(1)>HN(2) THEN A$="YOU" ELSE A$=" I ":DL=2
1800PRINT TAB(9,5):A$;" win and deal first."
1810PROCDEL(400):PROCCCD(11):PROCCD(18)
1820CLS
1830ENDPROC
1900DEFPROCCA
1910DD=DD-1:F=RND(DD)
1920C=CC(F):CC(F)=CC(DD)
1930SS=(C-1) DIV 13+1
1940NN=(C-1) MOD 13+1
1950ENDPROC
2000DEFPROCCB(N)
2010LOCAL X,Y
2020X=CP(N) MOD 40
2030Y=CP(N) DIV 40
2040PRINT TAB(X,Y);CHR$(226)CHR$(230)CHR$(230)CHR$(228)
2050FOR J=1 TO 3:PRINT TAB(X,Y+J);CHR$(231);" ";CHR$(231):NEXT
2060PRINT TAB(X,Y+4);CHR$(227)CHR$(230)CHR$(230)CHR$(229)
2070ENDPROC
2100DEFPROCCC(N)
2110LOCAL X,Y
2120X=CP(N) MOD 40
2130Y=CP(N) DIV 40
2140D$=CHR$(232)+CHR$(232)
2150FOR J=1 TO 3:PRINT TAB(X+1,Y+J);D$:NEXT
2160ENDPROC
2200DEFPROCCD(N)
2210LOCAL X,Y
2220X=CP(N) MOD 40
2230Y=CP(N) DIV 40
2240D$="   "
2250FOR J=0 TO 4:PRINT TAB(X,Y+J);D$:NEXT
2260ENDPROC
```

Listing 3a: cut for deal

```
2300DEFPROCCD
2310D$=CHR$(224)+CHR$(224)+CHR$(224)+CHR$(224)+CHR$(224)+" "
2320PRINT TAB(3,3);D$;D$;D$;D$;D$;D$
2330PRINT TAB(3,4);D$;D$;D$;D$;D$;D$
2340PRINT TAB(3,6);D$;D$;D$;D$;D$;D$
2350PRINT TAB(3,7);D$;D$;D$;D$;D$;D$
2360MOVE 72,943:DRAW 1228,943
2370DRAW 1228,847:DRAW 72,847
2380DRAW 72,752:DRAW 1228,751
2390MOVE 72,943:DRAW 72,847
2400MOVE 1228,847:DRAW 1228,751
```

Listing 4a: display frame

```
2410PRINT TAB(2,1);"MY SCORE"
2420PRINT TAB(2,9);"YOUR SCORE"
2430PRINT TAB(32,9);"TURN UP"
2440PROCE
2450ENDPROC
2500DEFPROCE
2510PRINT TAB(2,11);"MY  ";"'";"  HAND"
2520PRINT TAB(2,16);"TABLE  ";"'";"  TOTAL"
2530PRINT TAB(2,22);"YOUR";"'";"  HAND"
2540PRINT TAB(34,1);"DEAL"
2550ENDPROC
```

Listing 1b: title

```
1000 MODE 0:LOCATE 3,12
1010 PRINT "CRIBBAGE PLAYER"
1020 GOSUB 1200:M=250
1030 GOSUB 1100:MODE 1
1040 WINDOW #1,1,40,25,25
1050 INK 0,2:INK 1,24:INK 2,0:INK 3,6
1060 BORDER 2
1070 RETURN
1100 FOR Z=1 TO 8*M:NEXT:RETURN
```

Listing 2b: initialise

```
1200 DIM VN(13),N$(13),CP(21),CC(52)
1210 FOR J=1 TO 13:READ N$(J),VN(J):NEXT
1220 DATA A,1,2,2,3,3,4,4,5,5,6,6,7,7
1230 DATA 8,8,9,9,T,10,J,10,Q,10,K,10
1240 FOR J=1 TO 4:READ S$(J):READ H:SG$(J)=CHR$(H):NEXT
1250 DATA H,228,S,229,C,226,D,227
1260 FOR J=1 TO 21:READ CP(J):NEXT
1270 DATA 608,612,616,620,624,628,632
1280 DATA 636,808,812,816,820,824,828
1290 DATA 435,408,412,416,420,424,428
1300 FOR J=1 TO 8:READ ZH$(J):NEXT
1310 DATA 15,31,A PAIR,A RUN,3 OF A KIND
1320 DATA A GO,LAST CARD,4 OF A KIND
1330 RANDOMIZE TIME
1340 W$(1)="YOU":W$(2)="I"
1350 RETURN
```

Listing 3b: cut for deal

```
1400 LOCATE 5,3
1410 PRINT"C U T T I N G   F O R   D E A L"
1420 DD=53:FOR A=1 TO 52:CC(A)=A:NEXT
1430 FOR J=1 TO 2:GOSUB 1700
1440 HN(J)=NN:HS(J)=SS:NEXT
1450 LOCATE 5,10:PRINT"My card"
1460 N=18:GOSUB 1800:GOSUB 1900
1470 LOCATE 5,20:PRINT "Your card"
1480 N=11:GOSUB 1800:GOSUB 1900
1490 M=100:GOSUB 1100
1500 N=18:GOSUB 2000:GOSUB 1800
1510 SUIT=HS(1):DEN=HN(1):GOSUB 2100
1520 M=100:GOSUB 1100
1530 N=11:GOSUB 2000:GOSUB 1800
1540 SUIT=HS(2):DEN=HN(2):GOSUB 2100
1550 M=200:GOSUB 1100
1560 IF HN(1)<>HN(2) THEN 1590
1570 LOCATE 12,5:PRINT"We'll cut again."
1580 M=300:GOSUB 1100:GOTO 1430
1590 IF HN(1)>HN(2) THEN A$="You":DL=2 ELSE A$="I":DL=1
1600 LOCATE 9,5:PRINT A$;" win and deal first."
1610 M=400:GOSUB 1100
1620 CLS:RETURN
1700 DD=DD-1:F%=RND%DD:IF F%=0 THEN 1700
1710 C=CC(F%):CC(F%)=CC(DD)
1720 SS=(C-1)\13 +1
1730 NN=(C-1)MOD 13 +1
1740 RETURN
1800 X=CP(N) MOD 40
1810 Y=CP(N)\40
1820 LOCATE X,Y
1830 PRINT CHR$(150)CHR$(154)CHR$(154)CHR$(154)CHR$(156)
1840 FOR J=1 TO 3:LOCATE X,Y+J
1850 PRINT CHR$(149);"  ";CHR$(149):NEXT
1860 LOCATE X,Y+4:PRINT CHR$(147)CHR$(154)CHR$(154)CHR$(154)CHR$(153)
1870 RETURN
1900 X=CP(N) MOD 40
1910 Y=CP(N)\40
1920 D$=CHR$(203)+CHR$(203)
1930 FOR J=1 TO 3:LOCATE X+1,Y+J
1940 PRINT D$:NEXT
1950 RETURN
2000 X=CP(N) MOD 40
2010 Y=CP(N)\40
2020 D$="    "
2030 FOR J=0 TO 4:LOCATE X,Y+J
2040 PRINT D$:NEXT
2050 RETURN
2100 IF SUIT=2 OR SUIT=3 THEN PEN 2 ELSE PEN 3
2110 LOCATE X+2,Y+3:PRINT SG$(SUIT)
2120 LOCATE X+1,Y+1:PRINT N$(DEN)
2130 PEN 1
2140 RETURN
```

Listing 4b: display frame

```
2300 D$=CHR$(144)+CHR$(144)+CHR$(144)+CHR$(144)+CHR$(144)+" "
2310 LOCATE 3,3:PRINT D$;D$;D$;D$;D$;D$
2320 LOCATE 3,4:PRINT D$;D$;D$;D$;D$;D$
2330 LOCATE 3,6:PRINT D$;D$;D$;D$;D$;D$
2340 LOCATE 3,7:PRINT D$;D$;D$;D$;D$;D$
2350 PA%=24:PB%=602:PC%=376:PD%=328:PE%=280
2360 MOVE PA%,PC%:DRAW PB%,PC%,1
2370 DRAW PB%,PD%:DRAW PA%,PD%
2380 MOVE PA%,PC%:DRAW PA%,PE%
2390 DRAW PB%,PE%:DRAW PB%,PD%
2400 LOCATE 2,1:PRINT"MY SCORE"
2410 LOCATE 2,9:PRINT"YOUR SCORE"
2420 LOCATE 32,9:PRINT"TURN UP"
2430 LOCATE 2,11:PRINT"MY  ":PRINT" HAND"
2440 LOCATE 2,16:PRINT"TABLE":PRINT " TOTAL"
2450 LOCATE 2,22:PRINT"YOUR":PRINT" HAND"
2460 LOCATE 35,1:PRINT"DEAL"
2470 RETURN
```

READERS

HELP US TO HELP YOU:

MENTION

COMPUTING TODAY

WHEN RESPONDING TO ADVERTISEMENTS

OVER 140 AMSTRAD CASSETTE TITLES IN STOCK

OVER 90 *NOW* AVAILABLE ON DISK

CPM SOFTWARE

Macro 80, Microsoft Basic, Microsoft Basic Compiler, Turbo Pascal, BBC Basic (Z80), Purchase Ledger, Payroll, Database, other titles on request.

● TAPE TO DISC TRANSFERS ●

HARDWARE

Printers, Speech Synthesisers, CPC464 3" Disc, Timatic 5¼" 2nd Disc Drive also available.

RS232 INTERFACES INCLUDE

Sideways Rom's, Optional upgrades to BBC Compatible User port. Also available Arnors Maxam Assembler and Quma's Assembler.

Mail order welcome. Please send sae for full list to:

TIMATIC SYSTEMS LTD

Registered Office:
NEWGATE LANE
FAREHAM, HANTS PO14 1AN
Tel: FAREHAM (0329) 239953

Sales and Repairs:
FAREHAM MARKET
FAREHAM, HANTS
Tel: FAREHAM (0329) 236727

It's easy to complain about advertisements. But which ones?

Every week millions of advertisements appear in print, on posters or in the cinema.

Most of them comply with the rules contained in the British Code of Advertising Practice.

But some of them break the rules and warrant your complaints.

If you're not sure about which ones they are however, drop us a line and we'll send you an abridged copy of the Advertising Code.

Then, if an advertisement bothers you, you'll be justified in bothering us.

The Advertising Standards Authority.
If an advertisement is wrong, we're here to put it right.

ASA Ltd, Dept 2 Brook House, Torrington Place, London WC1E 7HN

This space is donated in the interests of high standards of advertising.

SPECIAL OFFER

3½" FUJI Microdisks

SS/DD £29.95
DS/DD £39.95

5¼" FLOPPY DISKS

Suitable for use on nearly all single and double-sided 5¼" disk systems.
- Replacement Guarantee
- Hub rings
- Boxed in 10s

PRICES PER BOX

	1-4	5-9	10+
DS/DD	9.90	9.00	8.60

PRICES PER BOX

	1-4	5-9	10+
SS	29.95	29.50	29.10
DS	39.95	39.30	38.70

Full Lifetime Guarantee

99P EACH
(Boxed in 10s)
EXCL VAT

REPLACEMENT GUARANTEE
If any disk should fail, return it for free replacement. .

SEE10 LIBRARY CASES
(Hold 10 disks)

	1-4	5-9	10+
5¼"	1.99	1.85	1.75
3½"	1.99	1.85	1.75

DELIVERY

	1-4	2-4	5+
Disks (per box)	75p	50p	40p
Library Cases (Free with disks)	60p	30p	25p

HOW TO ORDER

To total order value add Delivery, then add 15% VAT and send to:

IDS Computer Supplies

P.O. BOX 436, BANCROFT, MILTON KEYNES MK13 0QX. Tel: (0908) 310896.

ALGORITHM ANGLES

Bill Home

Puzzle solving algorithms.

This month, we look at algorithms for puzzle solving. The puzzle is as follows:

Six men sat at a table, three with their backs to the window, three facing the window. We will call them A,B,C,D,E,F. The man opposite D was seven years older than the man on B's left, and five years older than the man on A's right. Both C's neighbours were younger (in years) then the man opposite E. F, who sat facing the window, sat opposite his twin brother. The ages of the three men facing the window totalled 100, and so did the ages of the three men with their backs to the window. At least two of the six were younger (in years) than the twins.

What were the men's ages, and where did they sit?

PRINCIPLES

Solving a puzzle of this kind is largely a matter of elimination. The given facts rule out some seating arrangements, and the facts must be converted to statements. However, the way the seats are numbered can be highly significant in this. It is desirable that a move one place to the left should increase the seat number by one, a move to the right decreasing the number, which leads to this plan;

WINDOW
1 2 3
6 5 4

Exploring all the posible arrangements is not easy. The method chosen here sets up 10,800 combinations, and then discards those in which the same seat numbers are given to two or more men. A mere 720 are left, and these are subjected to further scrutiny. This is not an efficient approach, but it is reasonably simple. Speed could be increased by putting the equality checks between the FOR statements, but speed is not very important for a one-off program.

THE RULES

Since there is someone on A's right, his seat is neither 1 nor 4. (Lines 120,210)

Since there is someone on B's left, his seat is neither 3 nor 6.(Lines 130,210)

Since C has two neighbours, he must be at 2 or 5.(Lines 140,220)

C's neighbours are younger than the man facing E, so E does not face either of them: C+E=6 and C+E=8 are excluded. (Line 220)

If B+D=6, the man opposite D is on B's left and cannot be seven years older than himself. Similarly A+D=8 is unacceptable.(Line 230)

If someone is on B's left and A's right, he is given two ages, so this is unacceptable.(Line 240)

F sits in 4,5 or 6.(Line 170)

These rules eliminate all but sixteen of the combinations. Those that remain need further examination.

The first objection arises when one line or the other has a total 3*X + 9, which cannot equal 100. This is detected by giving X an arbitrary value of 30, allotted to the man on B's left. This makes the man opposite D 37, and the man on A's right 32. F and the man facing him are made equal if either is zero, and the line totals are checked. If the result is 99, the word 'HUNDRED' marks an exclusion. If F and his opposite number have been defined as being of different ages, the word 'CLASH' marks another reason for exclusion.

For the remaining eight cases, the ages determined are printed out. When no age has been given to F or to the man facing him, the system fails, but the result remains useful. In the first combination, C's neighbours are both younger than the man opposite E, and the same applies to the fourth combination, but the rest are ruled out by the 'neighbour' requirement.

In the fourth combination, no one is younger than the twins. If you experiment with this layout, keeping A,B and E at the same relative spacing, you will find that no variant meets both the 'twin's age' and the 'neighbour' rules. We are left with the first combination, which satisfies all the requirements.

COMMENT

It is unusual to find that a puzzle of this type can be solved so completely by a computer program. It would have been impossible to signal the 'neighbour' exclusions automatically, by checking for either V(C+1) or V(C-1) being equal to or greater than V(E), while detection of the 'twin's age' exclusion might have been possible but would have entailed more calculation. The program would have been a bit of an anti-climax.

The problem with puzzle-solving is that it is liable to involve algebra sooner or later. In this case, the use of arbitrary numbers avoided this, without loss of generality.

The program was written for the AMSTRAD CPC464, and users of that machine will appreciate that an input of 0 at line 110 will give a displayed output, whereas an input of 8 will give an output on the printer. This conveniently allows a screen run to check for errors, after which the printout can be produced.

The program takes six minutes to run, but working out the solution manually could easily take six hours!

```
100 CLS
110 INPUT "PRINT";Y
120 FOR A=2 TO 6
130 FOR B=1 TO 5
140 FOR C=2 TO 5
150 FOR D=1 TO 6
160 FOR E=1 TO 6
170 FOR F=4 TO 6
180 IF A=B OR A=C OR A=D OR A=E OR A=F THEN 480
190 IF B=C OR B=D OR B=E OR B=F OR E=F THEN 480
200 IF C=D OR C=E OR C=F OR D=E OR D=F THEN 480
210 IF B=3 OR A=4 THEN 480
220 IF C=3 OR C=4 OR C+E=6 OR C+E=8 THEN 480
230 IF B+D=6 OR A+D=8 THEN 480
240 IF (B=1 AND A=3)OR(B=4 AND A=6) THEN 480
250 A$(A)="A":A$(B)="B":A$(C)="C"
260 A$(D)="D":A$(E)="E":A$(F)="F"
270 FOR X=1 TO 3:PRINT#Y,A$(X);:NEXT
280 PRINT#Y
290 FOR X=6 TO 4 STEP -1:PRINT#Y,A$(X);:NEXT
300 FOR X=1 TO 6:V(X)=0:NEXT
310 V(B+1)=30:V(7-D)=37:V(A-1)=32
320 IF V(7-F)=0 THEN V(7-F)=V(F)
330 IF V(F)=0 THEN V(F)=V(7-F)
340 IF V(7-F)<>V(F) THEN PRINT #Y,TAB(6);"CLASH":
GOTO 470
350 IF (V(1)+V(2)+V(3)=99) OR (V(4)+V(5)+V(6)=99)
THEN PRINT #Y,TAB(6);"HUNDRED":GOTO 470
360 IF V(1)=0 THEN V(1)=100-V(2)-V(3)
370 IF V(2)=0 THEN V(2)=100-V(1)-V(3)
380 IF V(3)=0 THEN V(3)=100-V(1)-V(2)
390 IF V(4)=0 THEN V(4)=100-V(5)-V(6)
400 IF V(5)=0 THEN V(5)=100-V(4)-V(6)
410 IF V(6)=0 THEN V(6)=100-V(4)-V(5)
420 PRINT#Y,TAB(5);
430 FOR X=1 TO 3:PRINT#Y,V(X);:NEXT
440 PRINT#Y
450 PRINT#Y,TAB(5);
460 FOR X=6 TO 4 STEP -1:PRINT#Y,V(X);:NEXT
470 PRINT#Y:PRINT#Y
480 NEXT:NEXT:NEXT:NEXT:NEXT:NEXT
```

THE ARNOR MAXAM

Bill Horne

A profusion of assemblers for the Amstrad CPC464 is now beginning to appear, but this one, from London-based Arnor, looks to be quite special.

Anyone who wants to do any serious programming needs at least a 'toolkit' and preferably an assembler. The necessary programs are usually loaded into RAM, reducing available workspace, which may already be at a premium. The Amor MAXAM program for the AMSTRAD CPC464 is available in ROM, leaving the already generous provision of free RAM virtually untouched. The Text Editor initially offers 41224 bytes of workspace, which should be enough for the most enthusiastic programmer. The only argument for using the alternative cassette or disc versions is that they are cheaper.

INSTALLATION

Mounting the tiny MAXAM unit is simplicity itself, once the functions of the three connectors have been sorted out. A small diagram would help here, though the instructions are clear enough if you read them carefully. The unit plugs into the 'disc drive' connector, the interface lines being brought through to a board-edge connector at the rear. The third connector provides for vertical entry of a further extension board.

The added unit tucks away neatly, partly inside the main computer case, so this is almost a 'fit and forget' situation, because there is no need to remove the unit after use (indeed, it is best left in place, to save wear on the connector surfaces.) It will sit there quite happily, coming into action only when required. At switch-on, the presence of the unit is reported in the initial display, but BASIC is in charge, as usual, and entry to MAXAM is obtained by a special command.

CONTROL HIERARCHY

MAXAM adds up to seventeen new command words, though some are not available in the disc and cassette versions (see table1). These take you from BASIC into the principal system functions. For example, MAXAM enters the main menu, which itself offers sixteen options, as listed in Table2. Type 'T', and the Text Editor menu appears, offering seventeen more options. This multi-level control hierarchy groups associated commands together, avoiding the need for a single massive menu.

One command allows a switch between Mode 1 and Mode 2. This text will be printed on an 80-column basis, so Mode 2 is in use — yellow on blue, as usual, giving a readable display. Mode 1 colours are fully exploited, giving a very pretty display, though the red is less readable than the other colours.

TEXT EDITOR

The Text Editor is not quite a word processor, though it is in use as such for writing this article! Line and column counts are displayed on the top screen line, together with the number of free bytes and the current working mode (i.e. Insert or Overwrite, selected by pressing Control and Tab.) This differs from some text editors associated with assemblers, since it is usual to impose a rigid tabulation pattern. The free format allowed in this case is useful. Tabs can be set to suit personal fancy.

Blocks of text can be defined by markers, providing means for partial printout, but the more decorative falals of a full-scale word processor are — not unreasonably — omitted.

Incidentally, the editor imposes no limit to the maximum length of a line, the display wrapping round by half a screen width if the right hand edge is passed. This sometimes needs watching, to avoid the 'Where the dickens did that come from?' syndrome, because text outside screen limits can still be printed.

Among the more esoteric characteristics of the editor, there is provision for generating BASIC programs that include assembler sections. These sections are marked as blocks, and a MODIFY function is called to change the block markers to quotes. A further call inserts line numbers, and the modified text can be saved to tape.

But we are getting a little ahead of ourselves. The Assembler must now be considered.

THE ASSEMBLER

In general, an assembler is an assembler is an assembler. For a given CPU, there is usually not much difference between one version and another. You set up a text file giving the mnemonics of the required instructions, with a definition of link points and data locations. The assembler

FACTSHEET:	ARNOR adaptor AD1 with MAXAM (Amstrad CPC464)
Price:	ROM version £59.90 Disc version £26.90 Tape version £13.50
Available from:	ARNOR Ltd. PO Box 619 London SE25 6JL

scans through the text and works out the address structure, then makes a second pass to set up the required code. MAXAM does that, and quite a lot more. As already indicated, it will allow a program to be framed in a mixture of BASIC and assembler code. The command ASSEMBLE calls the assembler into action, and assembly code lines begin with a single quote symbol, telling the BASIC interpreter to lay off and do no tokenisation. (The equivalent of REM). When a line without the single quote is reached, BASIC comes into action again.

If you don't want the assembler to report itself, you use ASSEM, which suppresses the text. This is called 'invisible assembly'.

One of the bugbears of mixing BASIC and machine code is the transfer of variables from one to the other. MAXAM solves this by GET and PUT. GET will find the value of a BASIC variable and pass it to the machine code program. PUT will set up a BASIC variable from a machine code value. Using these functions requires a little practice, but they are very useful once you get the idea. Lengthy sample programs in the Amor manual give practical illustrations.

The manual, by the way, is almost 50 pages in length, in A5 size, and includes a glossary, a summary of Z80 instructions, and notes on the special AMSTRAD interpretation of the RST instructions. There are places where the explanations might be clarified a little, but this is likely to create a slight delay, rather than ultimate frustration.

The system worked equally well with cassette or disc, though there was one slight worry regarding the disc case. The plug-in disc interface is normally stabilised by pressing against the rear of the main unit, and that is lost when the MAXAM unit is interposed, making the overall combination a little fragile. You need to be careful not to bend anything in these circumstances...

EXTRA COMMANDS

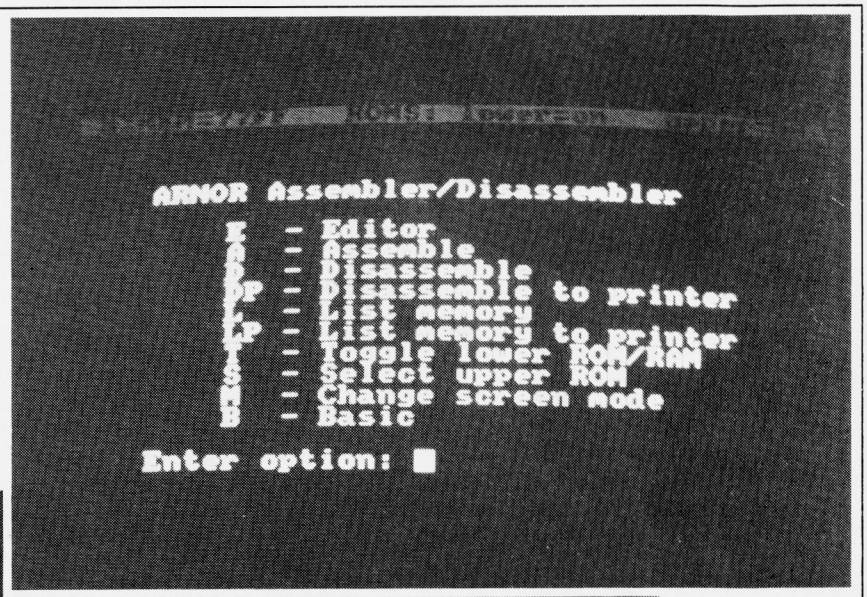
**	**MAXAM**	Enter main menu
**	**ASSEMBLE**	Enter assembler
**	**ASSEM**	Invisible assembly
**	**CLEAR**	Delete Editor text
*	**CAT**	Catalogue tape files
*	**FIND**	Locate string
*	**MODE**	Change between Mode 1 and Mode 2
*	**SPEED**	Set cassette speed
	HELP	ROM information
	HELP,n	Information on ROM n
	MAXOFF	Hide MAXAM
	MSL	Move screen memory low
	MSH	Move screen memory high
	ROMOFF	ROM disable

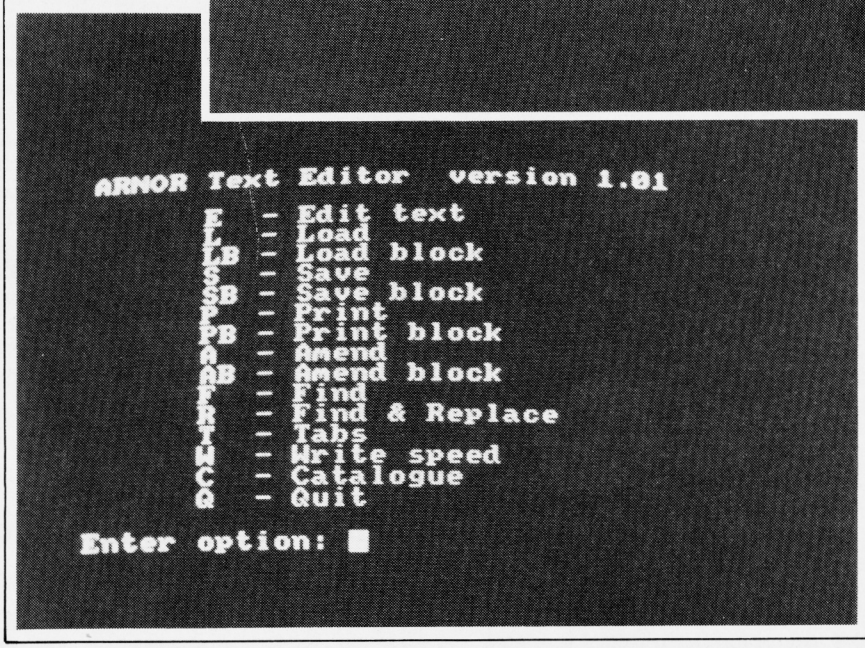
** Available on all versions
* ROM and disc versions only. Others available on ROM only.

MAIN MENU

T Text editor
D Disassemble
DP Disassemble to printer
L List Memory
LP List Memory to printer
S Select upper ROM
O Lower ROM ON/OFF
E Edit Memory
F Find string
FP Find string and print
M Move memory block
R Relocate block
I Initialise block
C Compare blocks
X External command
B BASIC

Right: The MAXAM main MENU





Left: options available from the MAXAM text editor

CONCLUSIONS

So what have we here? First, we have an ingenious use of the AMSTRAD ROM extension system, the first we have seen, though Melbourne Publishing in Australia have been experimenting in that area. This begins to approach the lost ideal of a system which will provide instant access to more than one main program. That, in itself, is worthy of note, but the innovation would be useless if the programs in the ROM failed to work. As far as we have been able to determine, they work very well, and bring to the CPC464 some of the capabilities which were adjudged to be missing. It would have been nice to have some provision for debugging programs, but one must not be greedy. In fact, the facilities offered would be difficult to match on any other machine of this general type. The CPC464 seems to be succeeding, and the addition of this add-on should increase its popularity.

TRAPPING ZEROES

R Maguire

Put an end to 'No room' errors on your BBC micro, with this useful routine

When an error occurs during the running of a program on the BBC micro, a message is printed on the screen indicating the error that generated the message. Pages 474 to 482 of the user guide gives explanations of these messages, and the error numbers associated with each.

To trap an error and so stop the premature ending of a program, a line such as ON ERROR GOTO (line number) can be inserted near the beginning of a progam. In which case, when an error occurs a GOTO (line number) statement is executed, and the program continues to run from that line.

A routine could be situated at this point to deal with the error, by using the number that is associated with that error. For example, a line such as IF ERR=17 PRINT "You pressed ESCAPE" could be used. After printing the message the program would continue from the next line in the program unless otherwise directed.

An approach like this works very well until we come up against ERR=0, which is the No room error number generated when the computer runs out of memory.

Ordinarily, error 0 is untrappable and hence program failure will occur. However, there is a way of anticipating when RAM is virtually exhausted.

Example 1 shows a small program which asks for strings to be entered. Strings are stored near the top of the computer's user RAM. After approximately 100 characters have been entered the program ends and a No room message appears on the screen. This indicates we have now used up all the available user memory.

Now insert lines 35 ON ERROR GOTO 100 and 100 IF ERR=0 PRINT "Out of memory". When the program is run again, exactly the same thing happens, as line 100 cannot handle this error. Try changing line 100 to IF ERR—17 PRINT" You pressed ESCAPE", and press ESCAPE before the program ends.

The answer to this problem is to constantly monitor the memory as it is been used up, then, just before we run out of memory, issue our own message so that the program is not terminated.

Fortunately, the computer's operating system keeps a record of the information we need, in its own dedicated RAM at locations &02 (low byte) and &03 (high byte). To see how these locations are updated, add line 65 PRINT ?&03; ?&02 to Example 1. It can now be seen that as each string is entered the amount of memory used is indicated on the screen. Remember that in MODE 7 the screen display uses memory from &7C00.

There is however one final problem to overcome before we can use the numbers held at &02 and &03 as indicators. On typing PRINT string$ (X-1) after the program has crashed, it will be seen that the last string$ entered can be lost. So we must protect against this eventuality.

To do this we must monitor individual characters as they are entered. Example 2 shows a method of achieving this by using the GET$ function. Line 50 sets up a buffer for storing the string as it is stored as a 20 character string and the program asks for the next to be entered.

In an actual, working, situation it would be better to use this feature to limit the length of the string, then return to enter the same string. The reason we might want to take this action is that failure to do so would require the programmer to leave a space of at least 240 characters before the beginning of the screen memory, to enable storage of the BBC's keyboard buffer string.

The input routine must now also check to see when the DELETE key is pressed, which is the purpose of lines 120 and 130

It is up to the individual user to decide what to do *after* trapping ERR=0 — Example 2 simply halts the program at the line 210.

Example 1

```
10 REM ** Example 1 **
20 REM ** Set PAGE to &7A00 **
30 REM ** before entering program.**
40 DIM string$(25):X=0:CLS
50 REPEAT
60 INPUT"Enter a string >"string$(X)
70 X=X+1
80 UNTIL X=25
90 STOP
```

Example 2

```
 10 REM ** Example 2 **
 20 REM ** Set PAGE to &7900 **
 30 REM ** before entering program. **
 40 DIM string$(25):X=0:CLS
 50 data$=STRING$(20,"*")
 60 REPEAT
 70 data$="":C=0
 80 PRINT"Enter a string >";
 90 REPEAT
100 G$=GET$
110 IF ?&03>&7A AND ?&02>&D0 THEN 210
120 IF G$=CHR$(&7F) AND C=0 THEN 100
130 IF G$=CHR$(&7F) AND C>0 THEN data$=LEFT$(data$,C-1):C=C-1:PRINT G$;
    :GOTO 100
140 PRINT G$;:data$=data$+G$
150 C=C+1
160 UNTIL G$=CHR$(&0D) OR C=20
170 IF C<20 THEN string$(X)=LEFT$(data$,C-1) ELSE string$(X)=data$
180 X=X+1:PRINT
190 UNTIL X=25
200 STOP
210 PRINT''"  Out of memory"
220 STOP
```


ACK DATA

For the BBC B Micro

BASIC COMPILER

This invaluable Software Package (containing around 60K of Code!) may be able to turn your BASIC programs into machine code, making them run up to 20 times faster! It comes complete with various utilities, including a disassembler and memory editor, and a detailed instruction manual.

CASSETTE	£26.95
DISK	£28.95

Send cheque/PO or SAE for details to

ACK Data
Dept. CT
21 Salcombe Drive
Redhill
Nottingham NG5 8JF
Tel: 0602 262498

BLANK DISKS

5¼in blank discs (40 track)
Prices per box of ten

	single-sided/ double-density	double-sided/ double-density
Scotch 3M	£14.95	£19.95
Dysan	£16.95	£24.95
Memorex	£12.95	£16.95

(unlabelled with free case)

S-J-B SUPER SAVERS!!!

50 Memorex SS/DD disks supplied in a perspex storage box.
only £59.95

50 Memorex DS/DD disks supplied in a perspex storage box
only £74.95
We also supply 3" and 3½" disks

FURTHER DISCOUNTS ON BULK ORDERS
CREDIT ACCOUNTS AVAILABLE
FREE FAST DELIVERY ON ALL ORDERS
ALL PRICES INCLUDE VAT

Please send cheques/PO's to:
S-J-B DISK SUPPLIES
DEPT CT 11 OUNDLE DRIVE,
WOLLATON PARK, NOTTINGHAM NG8 1BN
TELEPHONE: 0602 782310

POWERING UP

Gwen Nielson and Frank Brletich, Oneac Corporation

Advances in computer technology have altered the power requirements of modern computer systems and increased their sensitivity to power problems.

Up to 80% of all computer malfunctions and failures could be prevented if computer systems were provided with an electrical environment suitable for their sensitive components. Consequently, a comprehensive re-evaluation of the type and quality of power needed to operate modern systems is necesary before these failures and malfunctions can be effectively prevented and eliminated.

Two major changes have altered the computer's power needs. The increasing sensitivity of semiconductors and the move from linear to switching power supplies have resulted in systems that have much different power requirements than previous systems. As a result, tolerance to electrical noise has greatly decreased and systems that were once sensitive to small voltage variations now operate over a wide range of voltages as long as they are provided with a low-impedance source.

ELECTRICAL NOISE

The major cause of semiconductor failures in systems today is electrical noise. These excess power-line voltages exist at extremely high frequencies and can occur in nanosecond time frames. Depending on the energy levels, the impact on the semiconductor ranges from malfunctions to destruction.

Electrical noise has always created problems in computerized systems, but advanced production techniques have produced chips with a greatly reduced tolerance to noise. As densities increased and device real estate became more valuable, noise protection built into the chip was designed out. The result is much higher yeilds, but more fragile chips, so noise protection must be provided by the system in which the semiconductor is being used.

While most computer systems have some level of noise attenuation built in to prevent failure, the level is difficult to determine. Few manufacturers have determined the tolerances of their own systems, so end users should assume that a very minimal amount of attenuation is built into the system's power supply and look for a protection device that will ensure that only noise levels tolerable to the computer's delicate microcircuits are allowed through.

Noise levels over 0.5V can cause disruption of computer operation and result in garbled data or erroneous calculations, but normally will not result in failure. Degradation of a chip can start at just over 10V, creating a weak spot on the semiconductor surface. Once the degradation has started, heat is generated across the weak spot and the voltages experienced in everyday use will eventually lead to failure. Noise at approximately 40V can cause immediate semiconductor failure.

In 1980, IEEE Standard 587 was published. Based on the characteristics of building wiring, this standard states that a system can receive a pulse as severe as 6000V with a 0.5V sec risetime (Fig 1). Above this level the building wiring will arc over. A power conditioner designed to protect modern systems must be able to reduce this 6000V noise pulse to under 10V if semiconductor failure is to be prevented.

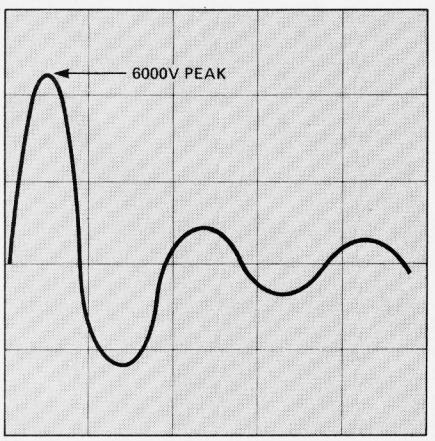


Figure 1: a system may receive a noise hit as severe as this 6000V, 0.5 sec rise-time 100KHz pulse. It is a 40% decaying waveform with a 200A current-source capability.

Noise occurs in two equally destructive forms: normal mode (line to line), and common mode (line to ground). Either type in levels over 10V can cause degradation system failure, and over 0.5V will cause disruption.

Common-mode noise generated across the safety ground line causes a special problem in today's computer systems because the safety ground is often used as the computer logic's reference point for communication purposes (Fig 2). The safety ground required by the National Electric Code to protect humans, and the reference point (system's common) have a coupling capacitance between them so that even if the logic circuits are floated in relation to the safety ground these two grounds are tied together at high frequencies. Normally, they are physically tied to earth through the utility ground. Water pipes and rods driven into the ground are also used.

Since the common-mode noise generated on the ground line has a direct connection to the semiconductors in the system, the quality of these grounds has a direct impact on system operations.

Protection devices that eliminate high voltages by dumping them into the ground line will actually generate common-mode noise levels whose magnitude depends on the transient voltage and on the impedance of the ground and neutral wires. Since power supplies provide a minimal amount of normal-mode attenuation and computer systems have no internal common-mode attenuation, the most effective product must reduce noise to under 10V normal-mode and under 0.5V common-mode

LINEAR AND SWITCHING SUPPLIES

The change from linear power supplies to switch-mode power supplies has resulted in modern computer systems having different operating characteristics than previous ones. Switching power supplies are generally used to increase efficiency and decrease size and cost in ac-to-dc power conversion.

Computers based on linear power supplies (Fig 3) are voltage-sensitive devices. Linear supplies maintain a stable dc output by converting the currents associated with excess voltage to heat through a dc linear regulator. Low-cost linear devices provide good regulation and fast response while providing moderate noise protection. However, they utilize a bulky 60-Hz input transformer and large heat sinks, so size and weight become problems.

Linear supplies also operate at low efficiencies (50% for a 5V supply), accept a very limited input-voltage range, and provide only minimal energy storage (less than 10 msec ride-through). And if the voltage level varies outside the supply's acceptable range, the device will shut the system down or, even worse, disrupt operations.

In contrast, most modern systems incorporate switching power supplies (Fig 4) which don't regulate the dc current directly. Instead, the pulse width modulates the incoming rectified ac current. Switching supplies operate at efficiencies of 70% or greater and have reduced size weight. They also provide energy storage in excess of 25 msec.

Most switch-mode supplies operate over a range of 90 to 135V because they regulate by adjusting the current pulse width. Thus, additional voltage regulation is no longer necessary. And rather than being voltage sensitive, switchers must be able to draw, on demand, the current required for proper operation. This requires a very low impedance power source.

The main disadvantage to the computer when you use a switching power supply is that switching at high frequencies (up to 100 KHz) is required to pulse-width modulate the current. Since noise is a high-frequency phenomenon, and the switcher functions at high frequencies, it will allow more noise to pass into the system than linear devices. To compensate, increased noise protection must be provided.

LOAD AND LINE CHARACTERISTICS

For a power conditioner to provide the proper interface between a computer and the utility, it must remain stable during operating changes in either system while eliminating power problems. A computer system is a dynamic load that must interact with an equally dynamic utility power line. As power -factor correction, grid switching, and the addition or reduction of loads occur throughout the day over the entire system, the power line goes from low to high impedance — and even the negative impedance — on a routine basis.

A computer system can interact with the utility during these changes without difficulty, but the power conditioner that provides the protective interface must be able to adjust to these changes without oscillation or overshoot. That is, it must be able to match the dynamic impedance of the line with the dynamic load characteristics of the computer system.

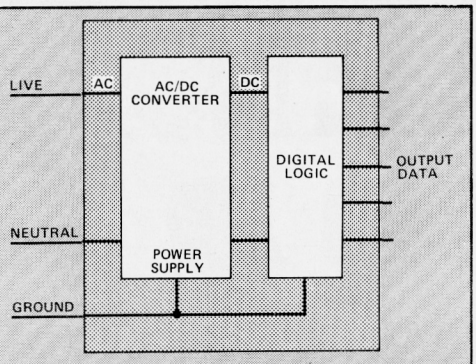


Figure 2: the digital logic in a computer system uses the safety ground as a reference. Common-mode noise over 0.5V generated on this ground line can cause system malfunctions and failure.

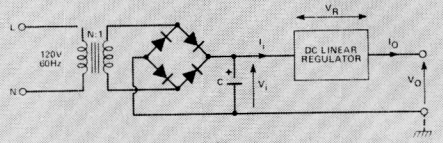


Figure 3: Computer systems based on linear power supplies are sensitive to variations in voltage level outside their design parameters.

MEETING NEW REQUIREMENTS

One power conditioner, developed by Oneac Corp. to meet the power requirements of modern systems, comprises three subsystems:

The input-filter subsystem represents the first line of defense against destructive transients by clipping their amplitude and slowing down the high-frequency components of the surge. In addition, it acts as an RFI filter, eliminating the extremely high frequencies associated with this type of noise.

The remaining attenuated noise passing the input filter then reaches the transformer section of the unit which must reject noise without introducing high impedance into the power path. Thus, proper current transfer is made to the computer's power supply without compromising noise rejection.

The third subsystem is the final elimination point for noise that may have passed through the other two subsystems. The virtual Kelvin ground ties line, neutral, and ground together at high frequencies. It eliminates any remaining noise but passes the lower frequencies necessary for power transfer. In addition, the low impedance design provides a power conditioner that meets the requirements of modern systems. Applying this type of interface to today's computer installations can improve system reliability up to 80%.

Frank Brletich, *general manager of Oneac Corp.'s Commercial Products Div., holds an MBA from Harvard Business School.*

Gwen Nielson, *marketing communications manager at Oneac Corp., has a degree in technical communications from Michigan Technological Univ.*

This article originally appeared in Winter 1983 edition of Computer Technology Review, and has been reprinted by permission of the publisher.

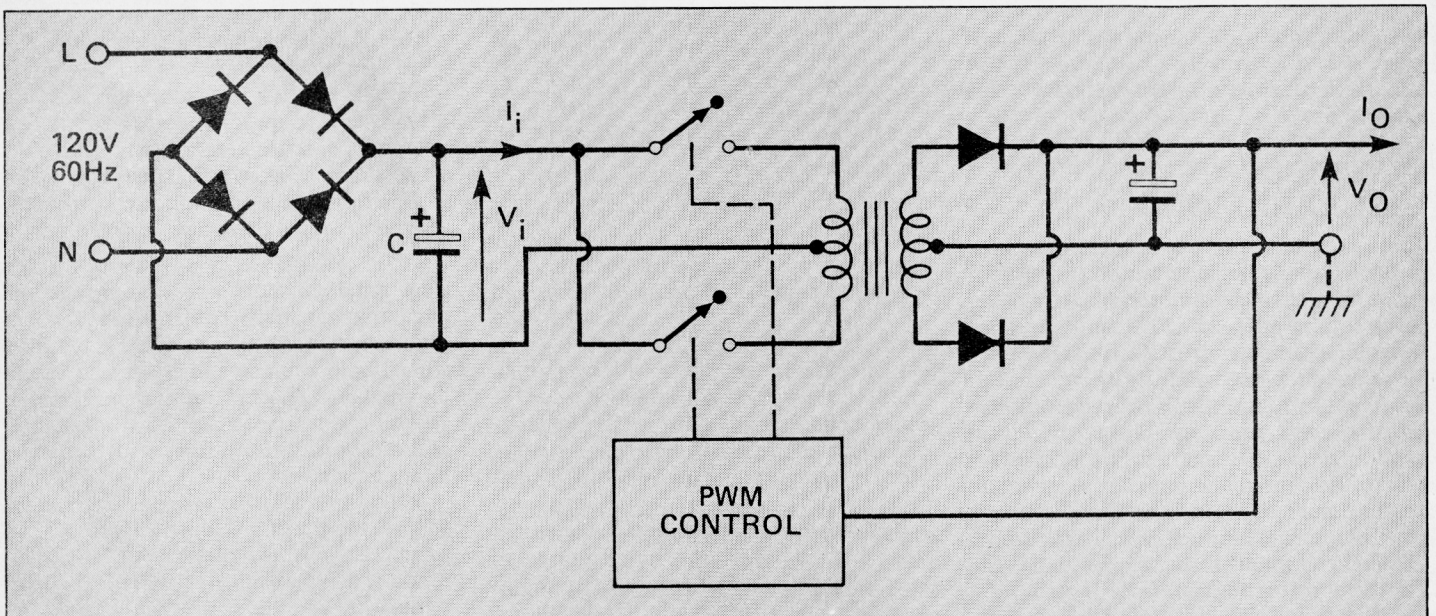


Figure 4: the switching power supplies used in most systems today are not affected by voltage variations, but must be able to draw current on demand from a low impedance source.

LEARN 'C'

Mark Woodley

Concluding our series on C, we discuss programming style and consider some do's and don't's worth bearing in mind.

This final article in the series, talks about programming style. Given here are a few general guidelines that should be followed when setting out your C programs. Let me stress that they are just that, *guidelines,* and that they may not apply to all situations. However, in the cases where these rules don't apply, they should be regarded as mere exceptions.

To start with, let's look at the layout of functions. The general way to set out a function was described in the first article and this is the way that the C compiler expects them. But, apart from satisfying the syntax requirements of a program, the C programmer can do much more to aid the readability of his programs.

VARIABLE DECLARATIONS

One standard, often quoted in C textbooks, is that variable declarations should appear in alphabetical order. However, by religiously following such a dictum, variables which are functionally related may appear in different places inside a function. For example, you may be writing a database management program. As well as having variables to hold each of the fields in a database record, the function may require other variables to hold disk addresses (sector and block numbers, etc.) as well as any other control variables which are particular to the operation of that function (e.g. Boolean variables, counters, accumulators). If these were to be declared in alphabetical order, then the functional relationships between the variables would be lost.

TYPE DECLARATIONS

It is a good idea to group type declarations in the order according to this standard:

```
extern
register
auto
static
```

(which, if you like, lists variables in order of *localness.)*

So, variable declarations which are related to each other by their use in the function should be declared together. Once this relationship has been made clear, then you can think about putting the variables into alphabetical order.

The same applies to definitions and type declarations. If a **#define** or **typedef** was particular to the operation of a function (as opposed to the operation of the program as a whole), then it should be made part of the function. If it was hidden away in some header file, it would be difficult to see how the program works, and at the same time, that function would be restricted for use only with that header file. Only those definitions and declarations common to the operation of **main** and the functions as a whole should be made in header files.

Use **#define** for constants, in particular those that may need to be varied later on, for example, buffer sizes. **typedef** should be used to simplify complicated declarations.

FUNCTIONS

In a similar vein, the functions in a program should be arranged so that they are related and this works well if a hierarchy chart (which shows the relationships between functions) is produced with the program, rather than listing them in alphabetical order.

Always declare the type of a function. Remember that each function is expected to return a value. Where a function does not return any value, the compiler should be informed by declaring the function as **void**.

Don't forget that **main** is a function and that its value can be returned to the operating system. In UNIX, this becomes the exit status of the invoking command.

It may be necessary to convert returned values to the appropriate type with the cast operator. So you will have to watch the type of a variable that is being 'return'ed.

The comments form an important and necessary part of each function. In general, there should always be an adequate comment at the beginning (usually after the argument declarations). It is seldom required to have additional comments, but where they are needed, they should appear to the right of the line to which they refer.

Functions should be kept as simple as possible. As you will learn with experience, it is a lot easier to write a program that consists of many short functions than to write a program that has two or three more involved ones. This works for several reasons:

IN WRITING short functions, the programmer is able to visualise most of the logic and coding in his head. This way, he is able to write it more quickly and more importantly, with improved accuracy. When the function has been written, it can be thoroughly tested on its own. Because there is less of it, any bugs inherent in the code will be much more obvious. Generally, in small sections of code, there are only a few paths to test.

AS a function grows, it becomes much harder to imagine. To make matters worse, in testing the function, there may be an infinite number of paths. As you might guess, the programmer who develops a program in large blocks, spends more time in writing it. The easy solution is to look for sections of code that could be written as separate functions, and delegate them down accordingly.

One principle that marks the top-limit of function complexity is derived from the fact that most programs become over-complicated when there are more than five levels of indentation. Other indicators of programmer overkill are functions that have more than half-a-dozen local variables or take up more than page to write.

In the same vein, expressions that appear within a program can often be improved. For example, you may quite often check the range of certain variables by some boolean expression:

```
('0' >= c) && (c <= '9')
```

This instantly becomes more readable if we adopt the mathematical standard in quoting ranges:

```
('0' <= c && c <= '9')
```

And because I don't want you to get into any bad habits I won't show you any nasty combinations of Boolean and arithmetic operators! As for the extrovert use of ? and :. Uugh!

Worse still, are uses of the scorned-upon keywords **goto** and **continue**. As Pascal successfully proved, it is quite easy to write programs without a single programmed jump, by keeping to the if..then, while..do and repeat..until structures. As far as the **continue** verb is concerned, its use only demonstrates that its programmer is only too willing to abandon a sinking ship!

C THE LANGUAGE

C is by its very nature an ambigious language to the newcomer. It is very easy to confuse the compiler by mixing the types of variables and by misunderstanding the precedence of operators.

Wherever possible, therefore, the programmer should use brackets in expressions to aid clarity, and always put braces around statements that are asked for by the syntax of a structured statement.

As an example, take the popular case of the nested if..else. If the compiler was given the following code:

```
if (expression)
    if (expression)
        statement1;
else
    statement2;
```

then although the programmer would have wanted statement2 to be dependent on the first **if,** the compiler would associate that statement and the **else** with the second **if.**

The situation can clearly be rectified by the use of braces:

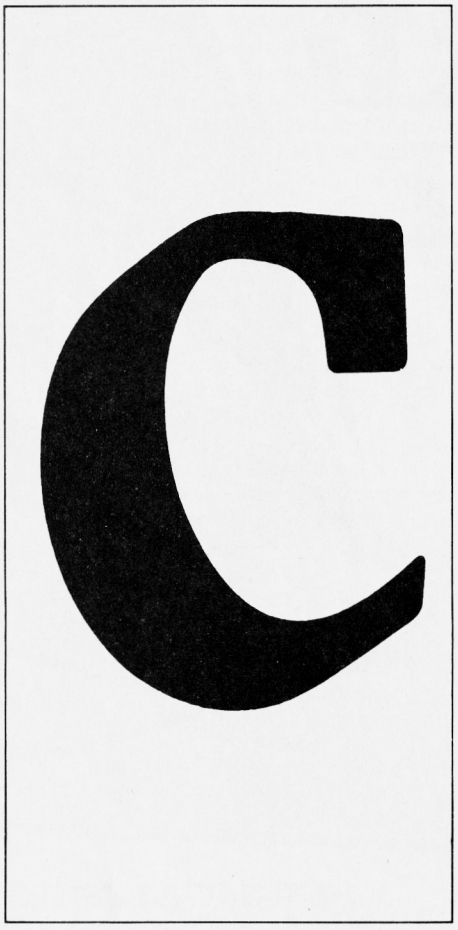
```
if (expression)
    {
        if (expression)
          statement;
    }
else
    {
        statement;
    }
```

This raised another important point. Make sure that the shape of the program reflects its actions. In the first example, this was not the case.

Other ambiguities can be raised by the use of #define. Bearing in mind that #define replaces every occurence of one set of characters with another, a line such as:

```
define    i    1
```

would replace every lower case **i** in the program with the character **1**, prior to compilation.



As you can imagine, this would have disastrous effects. For example, during compilation, every **if** statement would cause **syntax error**. This would no doubt cause much bewilderment to the poor programmer, and if it was well disguised, it may waste a great deal of his time in finding the cause.

This is one of the reasons why all predefined text in a C program is written in upper-case, with statements written in lower-case, so removing the possibility of such a traumatic event.

As an absolute top limit, the program as a whole should not be longer than 500 lines. If this is the case, then the program should be split into convenient (functionally related) parts, which are stored in separate files and **#included** at compile time.

Portability

Although C lets the programmer manipulate the machine at machine-code level, it is designed to be high enough so as not to actually dictate the hardware requirements. C was designed to be a portable language, so that the UNIX operating system could be implemented on a great variety of machines. The tremendous features of C's portability can be destroyed however, by bad programming.

At the beginning of this series I mentioned the representation of C's types. Not only does the amount of space allocated to types vary from machine to machine, but the way in which arithmetic types are encoded varies considerbly. A long integer, for example, may be stored as bytes 4321, 1234, or 3412 on different installations, where 21 is the least significant byte and 43 is the most significant byte. Therefore, the programmer should make no assumptions as to the representation of types. The only exception to this rule is in the use of arrays, where the ordering of types has already been described. Each element in the array can be accessed by supplying the appropriate element number, or by using a pointer that treats elements as a contiguous list. Because of this, declarations are designed to disguise the actual storage allocation provided to types, and so ease portability among machines with different data formats. Most C implementations are made very tolerant of inappropriate declarations and can pass values to and from functions and registers with no loss of accuracy. However, this tolerance, is not programmed into assignments as we have already discovered, and it is recommended that the programmer always ensures that he is passing items of the correct type.

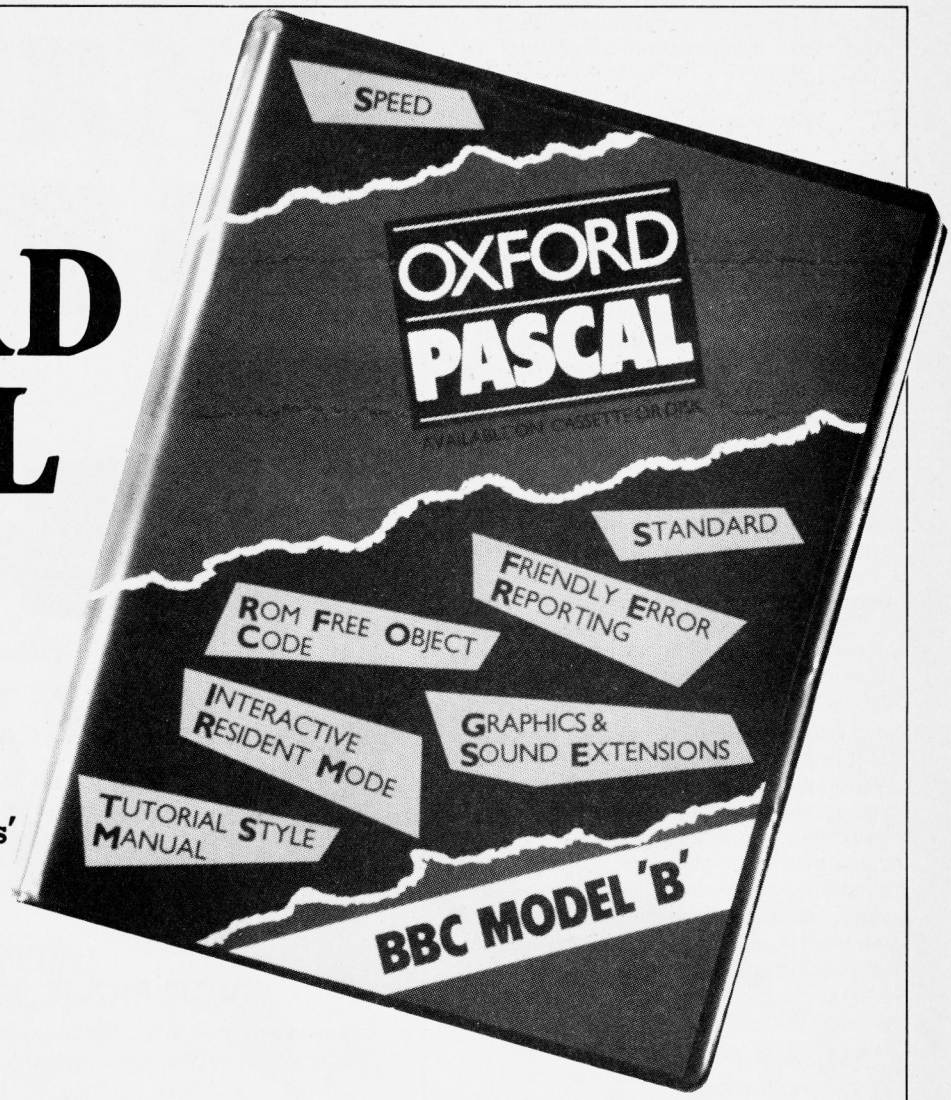
For example, the structured statements that involve expressions, expect them to be integers. Consider the **switch** statement. You might want to write some code like the telegram program, that inputs a stream of letters. For particular letters there may be a separate course of action and you want to select this with **switch**. If the expression used with **switch** is simply the **char** variable holding the current letter, then it will have to be cast to an integer. Otherwise, other bits might be misinterpreted by reading the character as an integer (e.g. sign bits) which would change the whole meaning of the program.

Never assume anything about the workings of the cast operator. Don't think that it will perform any sort of truncation, it won't, at least not below **integer.** Always assume that **short**, and **long** are distinct and different types on your machine. Even if they aren't, this may be the case on another machine. For this reason, beware of defining the size or bit patterns of constants.

Associated with your C implementation will be some manual explaining the system interface, how C works on your machine. Pretend that it doesn't exist. Don't even bother to discover the peculiarities of your host system (for example, the size of the input buffer). Not only will such practises destroy any portability in your programs, they may also change in later versions.

And finally, if you are using system calls for input/output operations, use those provided by the Standard Input/Output Library. These will invoke your operating system as C functions and will return values which you can check in case any errors arise. (For example, no diskette in the disk drive).

OXFORD PASCAL



Eraklis Pittas **looks at Oxford Computer Systems' Pascal for the BBC micro.**

The past 6 months have seen a positive surfeit of Pascal compliers being released for the BBC Micro. Each have been of varying degrees of quality and price. But the latest, and one of the cheapest on the scene, is Oxford Pascal from Oxford Computer Systems, who are probably better known for their implementation of Pascal on the Commodore 64.

Before describing the compiler, let's digress a little.

THE PASCAL LANGUAGE

Pascal was originally designed by Professor Niklaus Wirth at the Federal Institute of Technology in Zurich, in the late '60s. His definition of the language was subsequently published in 1971.

Wirth felt that the languages of the time were either too big and unwieldy on encouraged bad programming style. Pascal was therefore intended as an aid to teaching a systematic approach to computer programming — specifically *structured programming.* It was designed to be simple and elegant without being restrictive, by having a small number of powerful constructs, and the facility of user defined data types. This simple, elegant approach meant that the compiler was small, fast, and easy to transport to different processors.

ADVANTAGES OF PASCAL — OR OTHERWISE

The arguments over which language is best has, and will, continue for a long time. Suffice it to say that *no* language is better than all others for *every* application. Each language has its good and bad points, given a different problem.

When comparing BASIC and Pascal we see that the greatest attribute of BASIC is the ease of executing and debugging small-to-medium sized programs interactively. It is only when large complex programs/systems are tackled that Pascal (with certain extensions mentioned later) comes into its own. The usual arguments in favour of Pascal are more often than not about *structured programming.* This I feel is a white elephant, as you can write badly structured programs in *any* language. The opposite is also true; well structured programs can, and are, written in BASIC.

INSIDE THE OXFORD PACKAGE

For £49.95 (£39.95 for the cassette version) you receive a plastic case somewhat akin to a video-cassette case, consisting of a 16K ROM containing part of the compiler/editor system, the P-code interpreter and other library routines, a slim (95 page) well - produced manual, and either a cassette or diskette. The disk (in the case) contains the Pascal system that is external to the ROM, a run-time system (to run on machines without the ROM), a list of the Error messages that the compiler can emit, and 3 example programs; NIM, PATTERN and PERSIAN.

DOCUMENTATION

The manual, although slim, is clear and concise. It is split into two sections: a beginners guide to Pascal and a reference manual. Although the beginners guide is of the order of only 30 pages with a number of short list-

ings, it does serve its purpose as an introduction to Pascal, but should be supplemented by a *good* textbook. The reference section is no more than that, but for anyone who is not an absolute beginner the guide should serve as the only documentation needed to use the system. All the usual topics are covered: statements, data types, procedures and functions, pointers etc. A list of compiler error messages is included, so too is information on program chaining, and Pascal interfacing to BASIC and machine language programs. An index is included, but this only runs to a single page. Having said that, at least the manual was not a last-minute A4 'photocopied - sheets' job' which is usual with most of the less expensive compilers.

EDITING AND COMPILING

To use the system the supplied ROM must be put into one of the sideways ROM sockets — a procedure detailed in the manual. After this, getting into the system is as simple as executing a 'shift-break'.

After booting, the system is in the resident editor/compiler mode. In this mode the source text and the object code are both resident in memory. The editor supplied is not a full-screen editor but borrows from the built in BASIC line editor. All the standard BBC BASIC editor commands are present, with a few added extras:

DISC	Enter disc compiler mode.
RESIDENT	Reload resident compiler.
FIND	Find and print occurrences of a string in the program.
CHANGE	As find but substitute a second string for all occurrences.
PUT	Saves the program on diskette.
GET	Reads a program from diskette.

If the system is in the resident mode then a simple 'R' will compile the program currently in memory, and if there are no errors, will execute the resulting object code. An 'L' or 'P' will also compile, but will list the program on the VDU or printer respectively, as it does so. The resident mode is the only mode available to cassette users, and as such only programs of a few hundred lines can be compiled, which is ample for learning Pascal. If you have a disk then by entering disk mode the compiler is swapped-out and much more memory is available for program source. In disk mode, after a program is entered it *must* be PUT to disk, from which the command 'COMP <PROGNAME>' will compile the program into an object file called 'O.PROGNAME'. This file can be subsequently run with the command 'EX <PROGNAME>'.

All-in-all, editing and compiling programs is easy but would be easier if a full screen editor had been supplied, although the saving grace is that the format is identical to standard WORDWISE files.

PERFORMANCE

The object code produced by the compiler is not 6502 machine code, but PCODE: the code of a stack-based theoretical machine that is implemented in software (the PCODE interpreter). Although this code is interpreted, it is still faster than BASIC is all operations as far as I can tell, since it is in effect the machine code which is *fine tuned* for Pascal statement execution.

I ran a couple of my own benchmarks and found that straight FOR-NEXT loops ran about 50% faster again, but TEXT output to the screen was about 200% faster in Pascal! Although the tests were not by any means exhaustive, they served their purpose. When I ran the same tests on an Apple //e with Pascal 1.2 the Apple was from 30-100% faster in all the cases I tried, except in text I/O where the Beeb's faster hardware had the edge. This is interesting, as the BBC's processor runs at 2Mhz while the Apple's runs at 1Mhz — obviously there is room for improvement in the PCODE interpreter. Execution speed aside, the compiler itself is about as fast as the Apple Pascal compiler since it is written in Oxford Pascal itself and thus has to be interpreted.

FACTSHEET:	OCS Pascal
Price:	disc version £49.95 cassette version £39.95
Available from:	Oxford Computer Systems Limited Hensington Road Woodstock Oxford OX7 1JR Tel. 0993 812700

OTHER FACILITIES

Some very useful features which will be appreciated by professional programmers should be mentioned. Programs can CHAIN as others so that programs larger than the available memory can be split up, external source files can be included into a file that is being compiled and separate compilation is available, i.e. separate program object files or libraries of often-used Procedures or Functions can be linked together into one object file. This is a definite 'plus' to the system, and one that TURBO Pascal or Acomsoft's ISO Pascal does not allow. NEW and DISPOSE as defined by Jensen and Wirth is implemented, so too is the passiong of Functions and Procedures as parameters.

Finally, programs can be 'LOCATED' or compiled with a RUN-TIME library tacked onto the object file. This means that a program can run *without* the ROM being present. LOCATED programs can, therefore, be distributed or sold to users without the ROM. The best part of this is the liberal licensing policy of OCS, who simply ask for a mention in any documentation and makes no royalty or copyright claims on *your* software.

EXTENSIONS

Extensions to standard Pascal include all the BBC BASIC graphics commands. Also included are: SOUND, ENVELOPE, GET-KEY, VDU, OSVDU, OSCLI, HIMEN, SET-TIME, RANDOM, etc. In fact nothing has been overlooked. Hexadecimal constants are allowed in expressions and built-in Procedures to read and write HEX values are provided. Finally, for 'professional' programs, I/O errors and the <ESCAPE> key can be trapped and dealt with by the program.

BUGBEARS

No piece of software is perfect, and this package is no exception. A couple of extras could have made it infinitely better. Although extensions for reading STRINGS have been added to the ISO standard, it is nowhere near as useful as that of Apple (UCSD) Pascal — this should not be too difficult to add. Program segmentation is another easy addition that would be very useful. The only real ommission, though, is that Random Access files have not been provided. Although this is not part of the ISO Pascal standard, it really is a must for all serious Pascal compilers. The compiler is as close to the ISO Pasclal level 0 standard as makes no difference. The one major deviation that I noticed is that identifiers (variable names) are only significant in the first 8 characters; the standard says all characters should be significant. Finally, in the adverts no mention is made of the need for the BASIC 2 and OS 1.2 or greater to be present in your machine.

IN CONCLUSION

Oxford Pascal is good value for money with plenty of features, given the memory limitations of the BBC Micro. The excellent error messages given by the compiler — real English not error codes — would be very useful to someone learning Pascal. But this is not a toy compiler by any means. Even with the couple of gripes I had about random access files, very good commercial quality software can be developed with this Pascal. I can wholeheartedly recommend Oxford Pascal to Pascal novices and experts alike.

REFERENCE

Pascal User Manual and Report, third edition ISO Pascal Standard.

SUGGESTED READING

The above mentioned reference. (the bible fo Pascal).

Grogono, **Programming in Pascal** (A MUST).
1978, Addison-Wesley.

Bowles, **Problem Solving Using Pascal**.
1977, Springer-Verlag. (Uses TURTLE GRAPHICS as an aid)

Wirth, **Algorithms + Data Structure = Programs.**
1976, Prentice-Hall. (all programmers should have this).

TOWARDS THE Z800

Roman Iwanczuk, Zilog (UK) Ltd.

The preliminary specification for Zilog's Z800 family of microprocessors has been around for a long while, but customer samples are not yet ready. Here's a taste of the delights to come.

The Z800 family is a new generation of 16-bit, Z80 compatible, microprocessors which allow new designs to use the high performance features of an enhanced architecture. Z800 MPUs feature a comprehensive set of on-chip peripherals, and a 68 pin PGA 8-bit Z80 style bus and 16-bit Z-BUS interfaces which permit the Z800 microprocessors to offer a growth path for existing Z80 designs and a base for new high performance designs.

At the core of the Z800 microprocessors is an enhanced 16-bit version of the Z80 Central Processing Unit (CPU) which maintains the register structure of the Z80 but provides the Z800 with the extra dimension of flexibility. The Z800 retains the Z80 instruction set, maintaining object code compatibility, while incorporating both new addressing modes and new instructions giving the programmer more flexibility and power. Among the new instructions are a 16-bit signed or unsigned multiply and divide.

The Z800 CPU is based on a three stage pipelined architecture which, coupled with the on-chip 256 byte cache memory and a high clock speed, leads to a substantial increase in through-put. A further increase can be obtained by the Z800's programmable burst mode which supports "nibble mode" dynamic RAM's, effectively doubling the speed of bus accesses.

The Z800 also contains enhancements designed to aid it's use in multi-tasking operating system environments. These include a high performance, on-chip Memory Management Unit (MMU) and the facility for seperate system and user modes of operation with individual system and user address spaces. Separation of code and data in each address space is also a programmable option. The MMU allows dynamic logical-to-physical

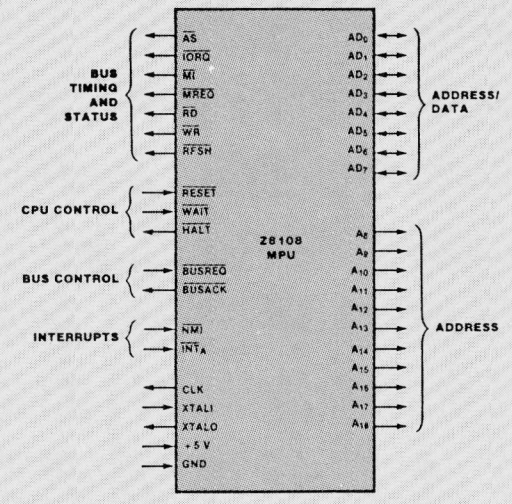


Figure 1. Z8108 Pin Functions

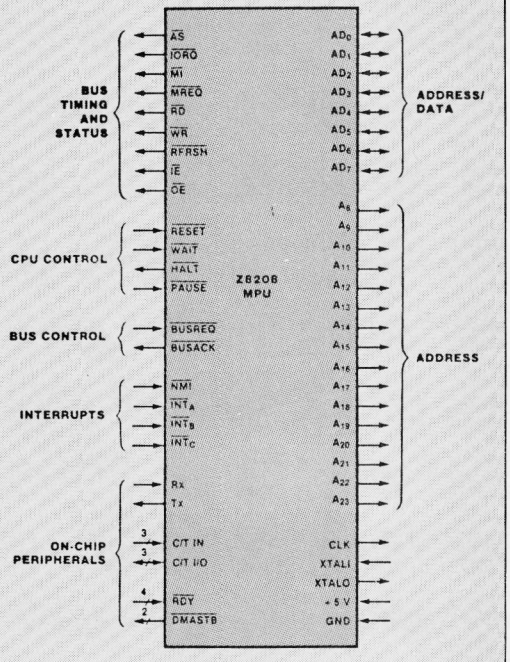


Figure 2. Z8208 Pin Functions

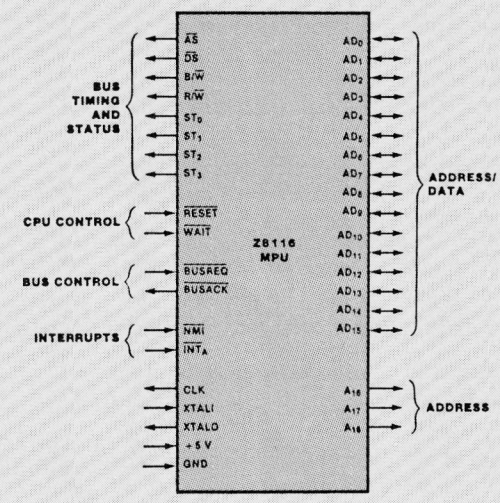


Figure 3. Z8116 Pin Functions

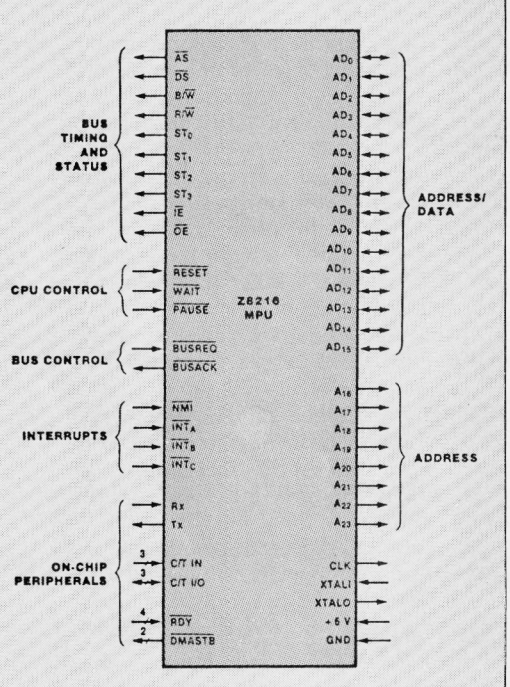


Figure 4. Z8216 Pin Functions

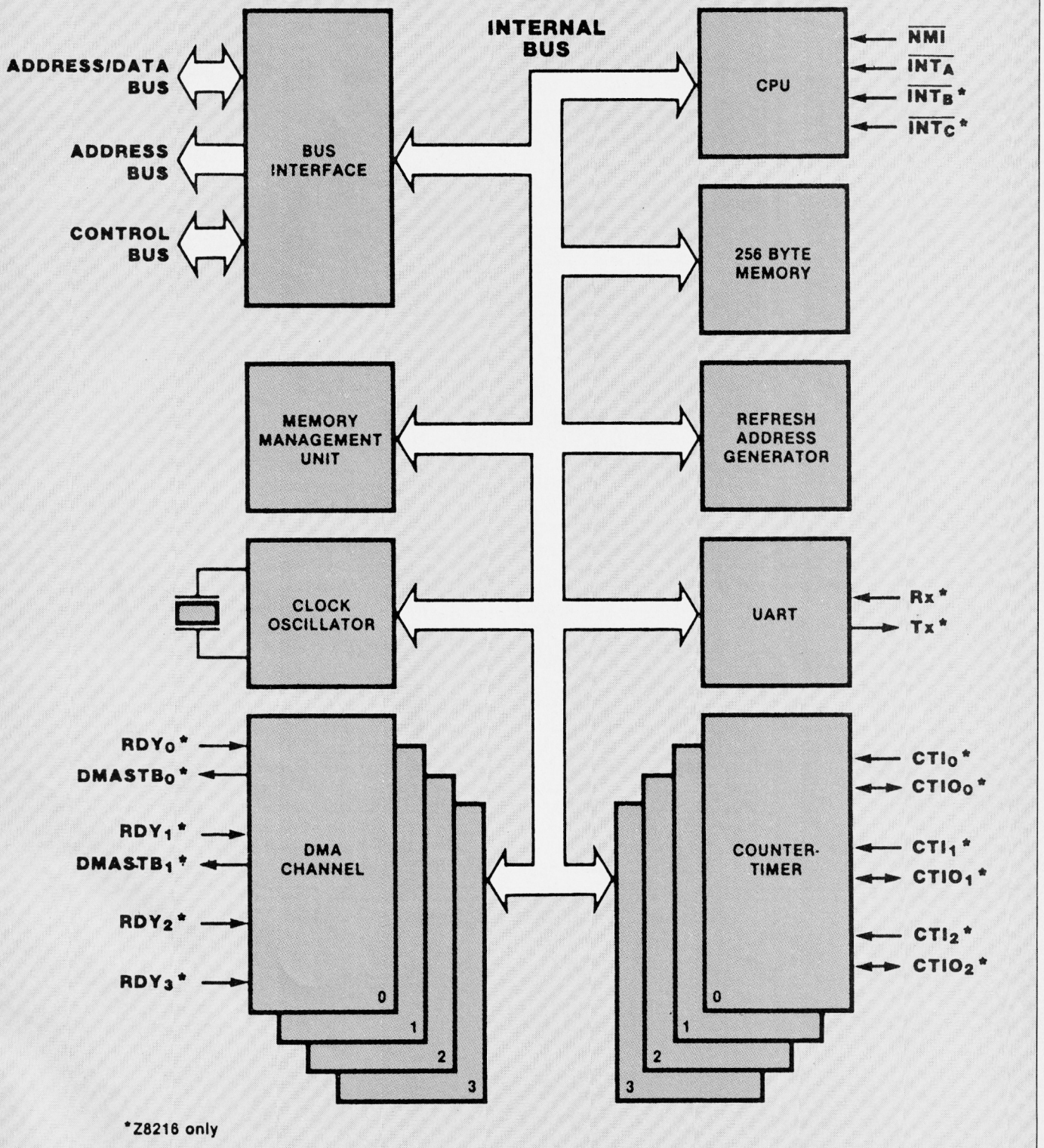


Figure 5. Z800 MPU Block Diagram

address translation, mapping a 64K byte logical address space into a 16M byte physical address space using 8K or 4K byte pages (depending on whether code/data separation is enabled or not). In addition, the MMU allows pages to be programmed as non-cacheable (barring information in a page from being copied into the on-chip cache memory during program execution) and/or write protected. These attributes are checked by the MMU when a page is accessed and, if an attribute violation is detected, the MMU inhibits the access, forces the CPU to abort the current instruction and generates an Access Violation Trap.

The Z800's I/O address space contains all of its programmable registers and on-chip peripherals. This does not impose a limit on the I/O addresses available for external peripherals because use of an I/O page register allows for generation of 24-bit I/O port addresses. The on-chip peripherals consist of Direct Memory Access (DMA) controllers, 16-bit Counter/Timers and a full duplex Universal Asynchronous Receiver/Transmitter (UART). Each DMA channel can be programmed for memory to memory, memory to peripheral, peripheral to memory or peripheral to peripheral transfers, and are capable of generating source and destination addresses which can be fixed, automatically incrementing or automatically decrementing. The Counter/Timers can be programmed as retriggerable or non-retriggerable and can also be programmed to use hardware or software gates and triggers. In timer mode a Counter/Timer acts as a down-counter, generating an external signal or interrupt on a zero count. In counter mode a Counter/Timer monitors an external input and decrements the down-counter on each low to high transition on the count input. Z800 family members without external Counter/Timer connections only support timer mode. The UART supports all common asynchronous formats and has additional features such as break detection and generation, error detection (parity, framing and overrun) and a programmable clock rate. Should additional peripherals be required the Z800 family will interface to Z80 family peripherals and to Zilog's family of Universal Peripherals. To further aid the design of peripheral interfaces, the Z80 family devices feature variable bus timing which allows the external bus speed to be 1, 1/2 or ¼ times the CPU speed and another programmable option allows 1, 2 or 3 wait states to be inserted during memory, I/O and interrupt acknowledgeing transactions. Additional enhancements that the Z800 provides are an on-chip crystal oscillator and a refresh addresses for dynamic memories with a programmable refresh rate.

The interrupt capabilities of the Z80 are all handled by the Z800 with the addition of an interrupt mode intended for use in operating system environments which further enhances the Z800's features. The Z800 also has three maskable interrupt inputs as well as a non maskable interrupt (only one maskable interrupt input is available on the 40 pin DIP version). In addition, the Z800 CPU supports eight internally generated traps which can inform the operating system of events such as System Stack overflows, MMU access violations or privileged instruction execution.

The Z800 family also supports Zilog's Extended Processor Architecture allowing the devices to interface directly to an Extended Processor Unit (EPU), such as the Z8070 Floating Point Unit, or trap EPU instructions so that software emulation can be performed. Z800 microprocessors support slave processors using a bus request/acknowledge protocol as well as multi-processor configurations with direct access to local memory and an external arbiter for access to a common global memory. The Z800 also supports multi-processor environments from the software point of view with a special Test and Set instruction.

Customer samples of the Z800 family will be available in early 1986, with technical manuals becoming available towards the end of 1985.

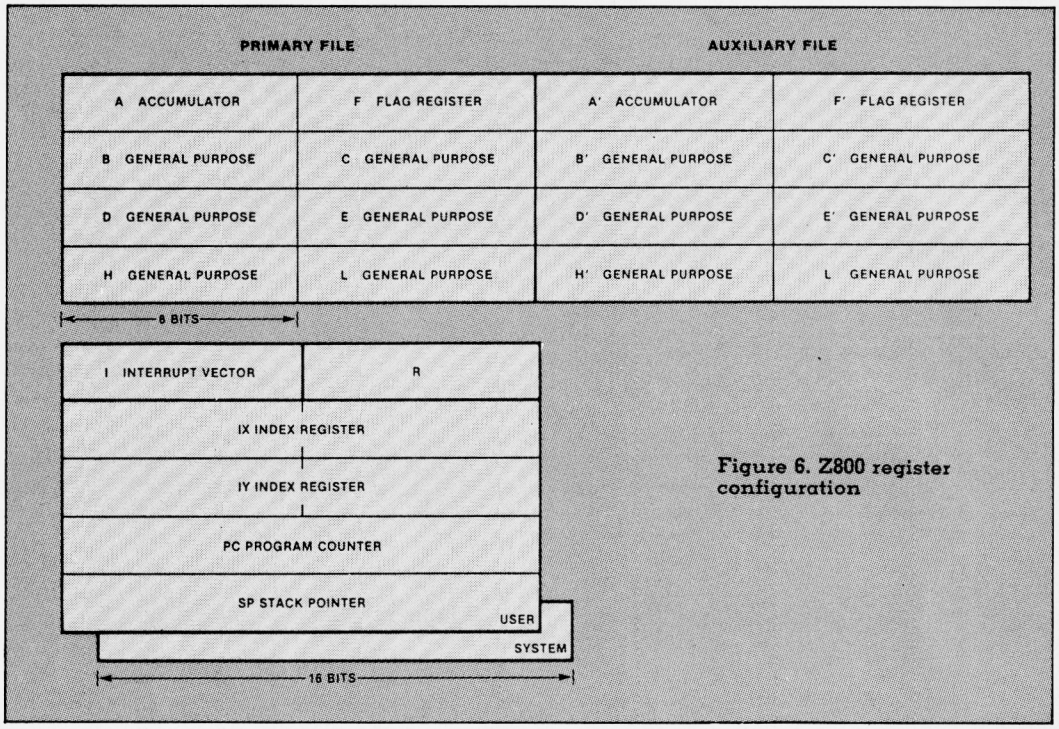
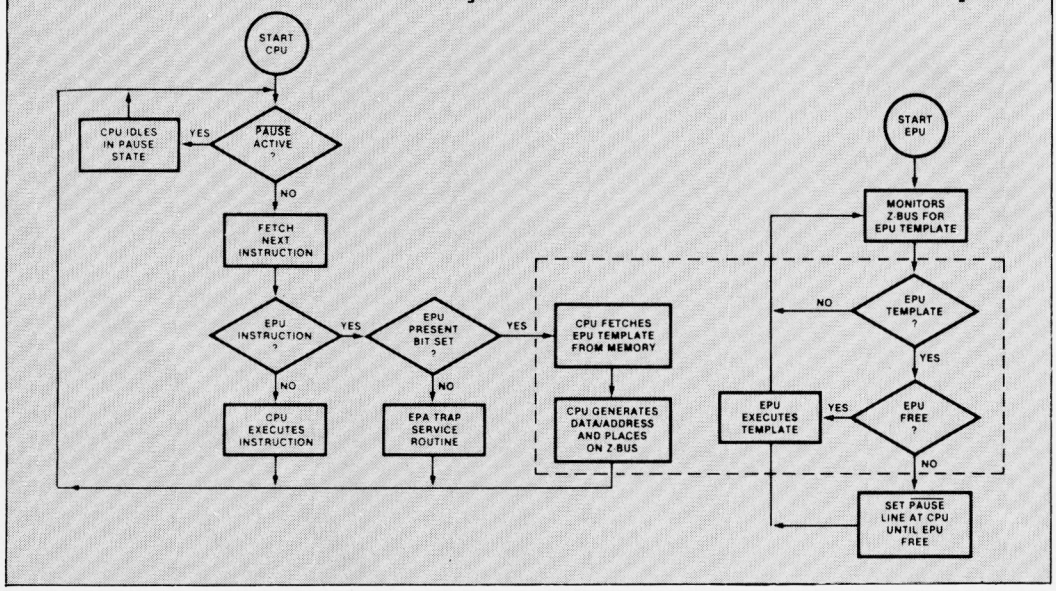


Figure 6. Z800 register configuration

Figure 7. EPA and Z8216 instruction execution cycle



SHORT SHARP SHOCK

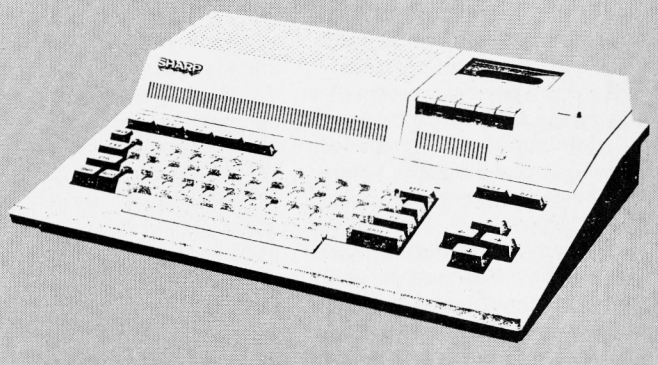
Stig Wetterling

Sharp appear to do a lot of business in eastern and northern European countries, with numerous variations on their British machines. One such variant - the Sharp MZ821 - has been brought to our attention, courtesy of a Swedish correspondent.

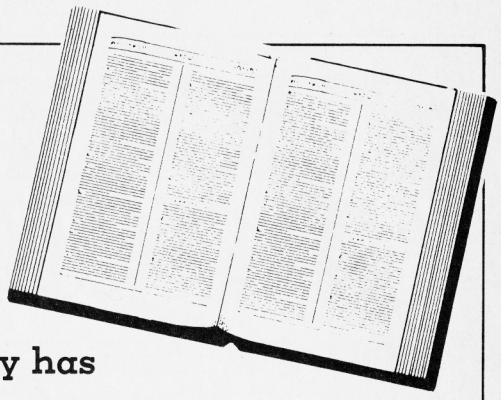
This isn't intended to be a full review of the Sharp MZ821, but it should offer just enough detail to hint at what Sharp is offering Europe, over and above the well known MZ80K etc.. I purchased my MZ 821 in Germany soon after its introduction last Autumn. It's price then was Dm1000 (£400 approx.), but now I understand it can be had for less than Dm1000 (£330 approx.).

The information here was obtained from Sharp electronics (Europe) GmbH in Hamburg, tel. 040 237751. The photostats give an approximate idea of what the machine looks like. Also given is a table 'Factsheet' style, listing the machines facilities.

FACTSHEET	**SHARP MZ821**
CPU	Z80A, 3.58 MHz.
RAM	64K, expandable to 128K
Video-RAM	16K, expandable to 32K
ROM	16K, containing boot routine, character generator, monitor, I/O routine, etc.
Display	RGB, Video and TV signals available, text 25 lines of 40/80 characters, graphics 320x200/640x200 pixels resolution,up to 16 colours, up to 4 screens adressable, soft scroll possible.
Keyboard	70 keys, ASCII standard, 2x5 programmable keys, cursor control keys, editor keys.
Sound	6 octaves, 3 channels.
Interfaces	Displays (see above), parallel printer (Sharp or Centronics compatible), plotter (Sharp only), cassette recorder, 2 joysticks, CPU bus, +5V for peripherals. 2 expansion slots (normally used for RAM expansion and serial interfaces or floppy disc controller).
Cassette	Built in, with automatic level and motor control, optionally replaceable with 2.8" Quick-Disk sequential microfloppy with 2x64K.
Options	64K RAM expansion 16K Video RAM expansion 128K Quick-Disk, see above Floppy disc controller for 5 ¼" floppies Single or double 5 ¼" floppy station P-CP/M operating system Serial interface board for modems etc. Sharp matrix printer Sharp 4-colour plotter Joysticks Monochrome monitor Colour monitor etc.
Features	MZ-700 mode selectable, makes the MZ 821 fully compatible with all MZ-700 software. Full screen editor built in. Extensive graphics included, with genuine high resolution. High speed BASIC, benchmark tests BM1 to BM7 being on par with BBC Model B, only BM8 taking approx. twice the time of the BBC. Large BASIC vocabulary, see separate list (153 commands). Documentation is reasonably good, the German version handbook is comprehensive, although several Basic commands have been omitted and a number of misprints are to be found. The keyboard is splendid; I just wait impatiently for a suitable text processing program to let loose on the 821. (This was written on my old and faithful Colour Genie!)



BOOK PAGE



Garry Marshall

Expert systems is our theme this issue, and Garry has found four very good books on the subject.

Over the past months several books on expert systems have come to *Computing Today* for review. I have kept them together and have been gamely struggling through them. I don't think that I'll ever be an expert on expert systems, but if you have the urge to be one, then between them, these books could be just the thing to get you started.

First of all, what is an expert system? It is a computer program that contains all the knowledge of an expert in some field and which can use it to do what the expert can. Further, it can perform just as well as the expert - and often better than an *individual* expert - in this role. To give two examples, expert systems exist that can successfully use their knowledge to detect mineral deposits from the data produced by prospecting and to diagnose particular types of illness by 'interviewing' patients to discover their symptoms. But it ought to be made clear at the outset that the expert systems that are operating successfully at present do so in very restricted areas. Systems for prospecting can find deposits if they are of a certain kind. Systems for diagnosis can diagnose illnesses if they fall in certain restricted category.

The way that expert systems have evolved to their present state (and they are still evolving rapidly) can be viewed from either of two equally instructive points of view. They can be seen as a natural development of conventional data processing, and they can be looked at as the most significant achievement to date of artificial intelligence. The first perspective may be the more comfortable, since most of us are likely to have background in conventional computing, and to know that expert systems have grown from this certainly makes them less daunting and easier to come to grips with. At the same time, everyone has heard of artificial intelligence, and it is worth knowing that, despite its detractors, it is not simply a theoretical subject but one can produce practical programs that are both useful and valuable.

With four books that are all more or less introductory treatments of a rather narrow topic, there is naturally a good deal of duplication of subject matter, although each has its own point of view and emphasis. Reading them all leaves an impression of four 'sliding panels' that can slide into the same positions to cover the same material, but which all slide out to different positions to cover different areas as well, so that although they duplicate each other they are also complementary.

QUESTIONS

Besides having to distinguish between the books, there is the extra problem of deciding how they complement each other. It is useful to have a few questions that we can ask of each book so that we can use the answers to help with both matters.

The answers to "What do we know about expert systems?" will allow us to assess what assumptions the books make about the readers' state of knowledge and allow us to judge at what point in the pile we should start. It is possible that we know nothing, or that we would like to begin as if we do. But I think it likely that most people will know at least a few things sufficiently well to feel able to build on them. These will probably include such things as that an expert system contains knowledge that has been extracted from an expert by a Knowledge Engineer; that knowledge consists of facts and of rules that can be expressed in the form of IF - THEN statements; that the raw data of facts and of rules does not become knowledge until it is structured and that its structure is usually tree-like; and that an expert system works by searching these trees using heuristics, or rules of thumb. It is probably also generally known that the computer languages Lisp and Prolog come into play when the programs for the systems are written.

The next question is "What do we want to know?" The answer to this can be anything between the two extremes that we always find in computing. At one extreme we have the users and potential users who only want to know what an expert system can do and how to use one once they have got their hands on it. At the other are the programmers who want to know how to write an expert system. I suppose that most of us will have an interest in the concerns of both extremes: finding the balance of these interests will determine the book that is most suitable. Incidentally, the people creating expert systems tend to refer to themselves as *engineers* rather than programmers. This reflects the image as the creators of products; also the certainty that, for anyone who is to feel comfortable in the information and knowledge-based industries, it must be ingrained that information is a raw material and knowledge is a product just as iron ore is raw material and a car a product.

Third is "What do we need to know?" Again, the answers can show considerable variety, but there are some basic things that we must know, and some slightly unexpected things that we shan't get far without knowing. Anyone with an interest in expert systems obviously wants to know exactly what it is, what it can do and how to use it. A knowledge of the principles underlying its construction is also necessary, that is, an understanding of how the system represents its knowledge and then processes it.

Whether we want to use an expert system or to create one, it is necessary to know the sort of areas in which an expert system can be applied to advantage. Rather surprisingly, it is possible to describe quite precisely the factors that characterise a suitable application area. It is also a good idea to have an appreciation of the limitations and shortcomings of an expert system, for these books make it abundantly clear that both do exist. There is a tendency to believe without question the output of a computer for no reason other than that it comes from a computer. But computer output should not be taken at face value indiscriminately. If garbage goes in then garbage comes out!

Buildings have fallen down because their designs were based on results taken without proper awareness from stress analysis programs. Providing the programs with inputs that were intended to deal with caused them to produce results, but results which were not valid. Similar things can happen with expert systems. And if an expert system for medical diagnosis were used incorrectly, a patient could die as a result. Of course, patients have been known to die even when treated by human experts, and this gives us a hint that it may not be so easy to assess the performance of an expert system even when we have what may appear to be a good one.

Our three questions should provide criteria for deciding which book to start with, and whether any book tells us not just what we want to know, but what we ought to know.

Before, at last, getting on to the books it seems a good idea to give some specific examples of existing expert systems with their areas of use and their achievements. The examples may help to crystallise the concept of an expert system. Additionally, the systems are referred to and described in all four books, and it helps to give the descriptions here so that I can refer back to them when dealing with a specific book.

PROSPECTOR is a system that acts as a geological advisor to geologists in the field. It can evaluate sites for the existance of certain deposits, evaluate the geological resources of a region and select the most likely sites for successful drilling. It's biggest success was in helping to find a deposit of molybdenum in Washington state worth an estimated $100 million.

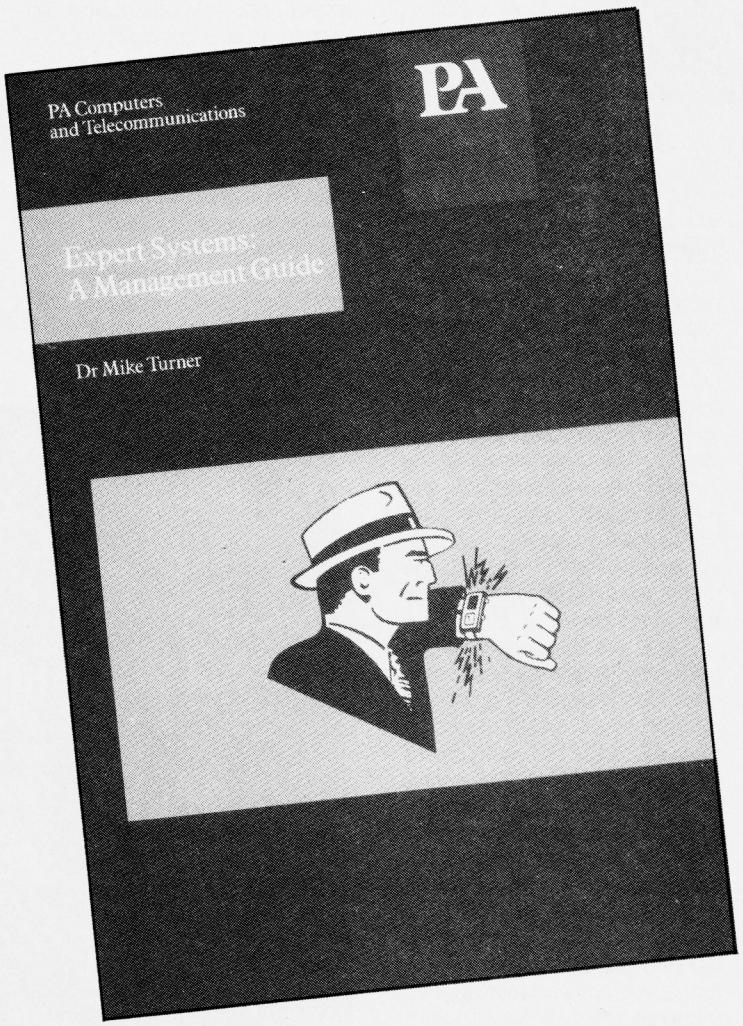
MYCIN is designed to help physicians by advising on the diagnosis and treatment of infectious diseases. It is interactive and if it does not have enough information to provide definite advice, it asks its user for the information it needs. It can also explain the reasoning behind its advice. Its scope is limited compared to that of a general practitioner but then, by definition, a general practitioner is not an expert.

DENDRAL was the pioneering expert system developed by a team led by Professor Edward Feigenbaum at a Stanford University. It infers the structure of chemical compounds from experimental data including that from mass spectrography and nuclear magnetic resonance.

XCON (formerly R1) deals with Digital Equipment's range of PDP and VAX minicomputer systems. It can determine if a configuration as ordered by a customer comprises a viable computer system and, if it does not, make it good. It can also connect the items in the configuration and to pack them in the company's standard cabinets. XCON's success rate is 99%, which is slightly better than that of human experts. Apparently, in its current state of development, XCON configures a system wrongly about as often as its human counterparts mistakenly think that it has.

Expert systems: a management guide' by Mike Turner is a zero-level entry to expert systems aimed at alerting managers to opportunities that they make available. This short book is all text: there are no equations or programs to be found here. It also provides all the information that a pure user of expert system could, and should, need.

Its overview of the subject begins by pointing out that expert systems are needed because human experts are scarce, expensive, busy, fallible and mortal. As a cheap and reliable way of having access to expertise, computer-based systems may be preferable to people.



The structure of a system is broken down into a knowledge base, a rule interpreter and interfaces to the user and to other sources of data. The knowledge base is explained as a collection of facts and rules expressing relationships between the facts. These rules are either absolute, as in:

IF you are a knowledge worker
AND you regularly perform knowledge-intensive tasks
THEN consider using an expert system

or they are fuzzy, as with:

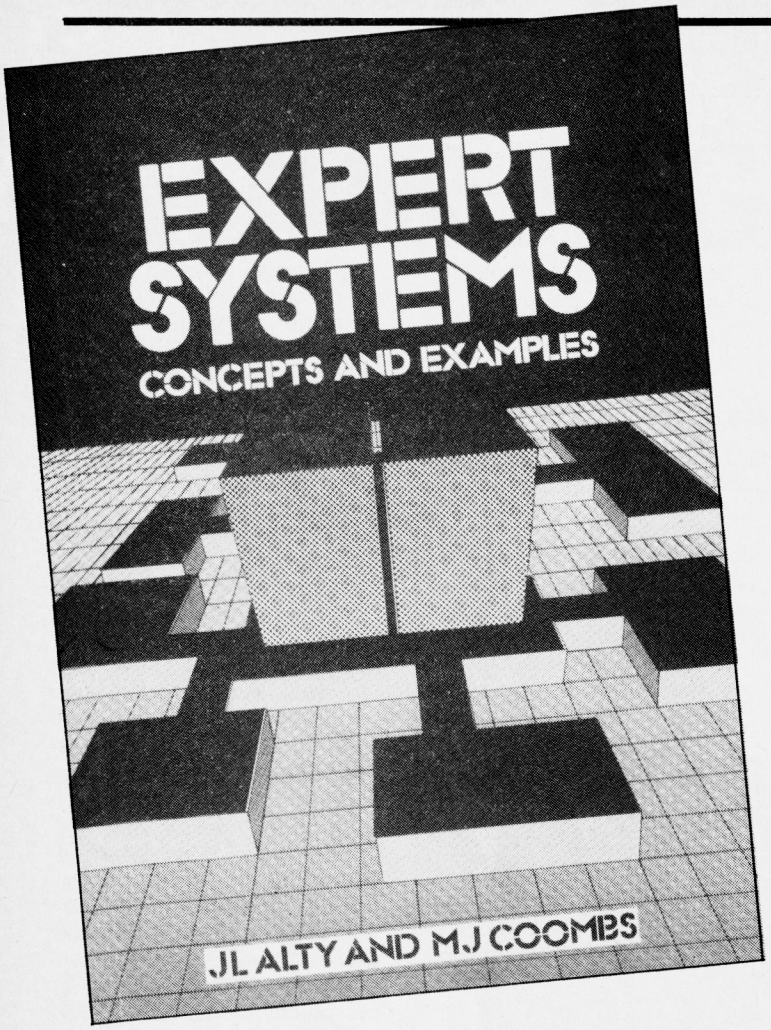
IF the applicant has a strong programming background
AND has extensive experience of real-time programming
AND the salary requirements are reasonable
THEN make a job offer

In the first example, the IF part involves propositions that are either true or false, so that their consequence follows as soon as they are tested. In the second one, things are not so simple, for we must not only tell the computer what we mean by 'strong', 'extensive' and 'reasonable' but also attach a scale of values to the interpretations. This shows that one way of representing knowledge can be dealt with by using traditional logic, while another requires a more general approach, which can be met by what is called 'fuzzy' logic. And the type of rule that has to be handled will depend on the application.

The Rule Interpreter is the way in which a system creates new knowledge from that existing in the knowledge base. We have just seen how new knowledge can be created by applying an IF - THEN rule, and the rule interpreter is essentially a strategy for controlling which rule is applied to which facts. It should not be hard to appreciate that some control is necessary, for without guidance the system will proceed blindly to try rules until it stumbles on one that works, taking goodness-knows how long to find one. By the same token, it should be apparent that different strategies are likely to be needed for tasks as different as diagnosing diseases and configuring computer systems.

The interface to the user allows the system to request more information from the user when it is needed. It also lets the user ask the system why it came to a particular conclusion. It is particularly important to the credibility of the system that it can explain its advice. In the Three-mile island incident, one of the computers controlling the power station initiated an unusual action which was overruled by one of the operators. Subsequently, it emerged that the computer's course of action, although unlikely, was correct. If the operator had been in a position to ask the computer to explain its action before he decided whether to abort it, the situation need not have arisen.

Systems can explain their conclusions with varying degrees of ease depending on how they work. A rule-based system need only recount the rules that it applied to generate its conclusion. Other systems cannot explain so easily. I have seen a system at Brunel University that could recognise the face of one of a small group of people when they entered its room, (It is certainly a system with some expertise.) It had been trained by exposure to the different faces on a television display, and it reacted to them by setting values on a number of potentiometers according to features of the display. It recognised a face by comparing the settings it produced with the settings remembered from its training period. The point is that



if you ask this system to explain how it recognises a face, all it can do is to give you a set of potentiometer readings, and this is not very illuminating.

APPLICATIONS

The author moves on to explain how to select the sort of applicant for which an expert system should be considered, and what the aims for such a project should be. All four books show a general agreement on the sort of area that is suitable. It should be one that cannot be done in any obvious way by conventional means, either because the data is uncertain or because it is computationally not feasible. The knowledge involved should be expressed more easily in rules than by equations or algorithms. It should be an area where advice and explanation is required, where the computer may have to be run by staff without specific computing skills, and where the knowledge in the system may be needed for use in more than one way, perhaps for training as well as advising.

The aims of such a project could include freeing human experts for other activities, supplying expertise that can easily be shared, managing complexity, ensuring a uniformity of approach to problem solving throughout a company, providing explanations for the courses of action that are recommended, gaining a competitive edge, making knowledge and expertise explicit, eliminating bias and prejudice, and many more.

After this, the feasibility of introducing an expert system to deal with a problem area in a company is discussed. A general discussion is followed by a case study of the introduction of ZCON to Digital Equipment. The conclusions of this case study are interesting because they show the benefits that can follow a successful introduction. XCON improved the service to customers in many ways, most notably by allowing orders to be checked more quickly and giving quicker deliveries. But the savings that result from its use in a year are about the same as its total developement costs.

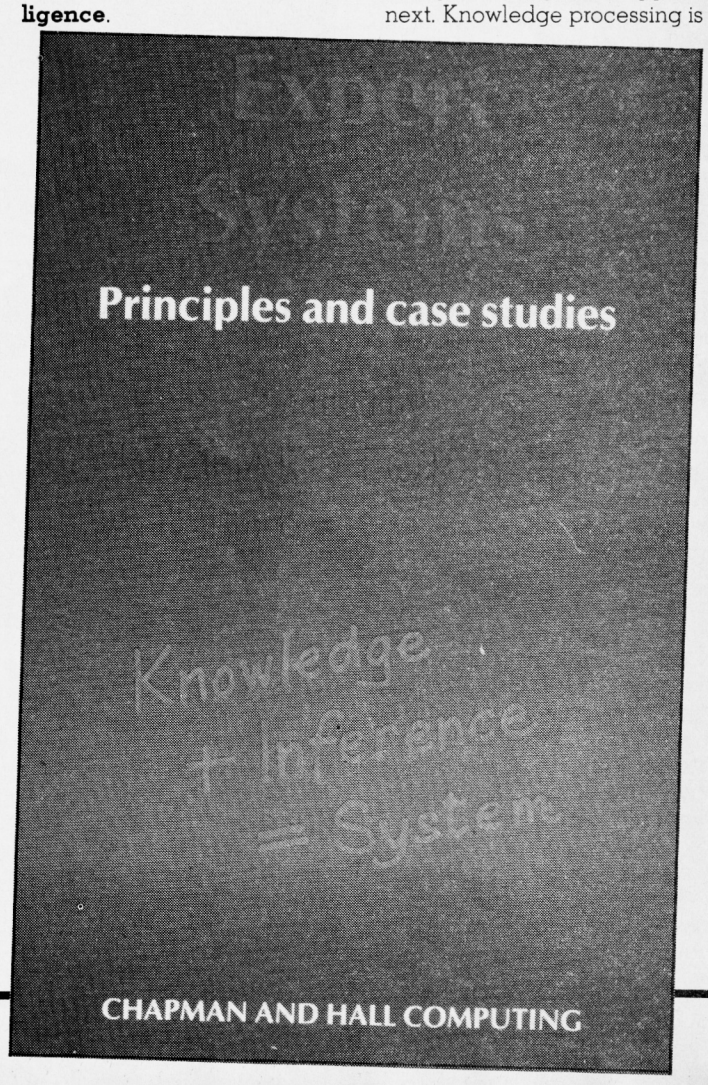
Finally, the book gives a list of potential applications in industry, banking office automation, defence, government and education which should provide an example relevant to the experience of almost any manager.

This book is a good starting point for anyone (manager or not) interested in using an expert system. Technical matters are relegated to two brief appendices, but its strength lies in its readable general account of why, how and for what an expert system should be used.

Expert systems: concepts and examples by J M Alty and M J Coombs is an introductory text, the first part of which explains the underlying concepts while the second supplies a selection of examples. The authors are from the University of Strathclyde, and their book is aimed at students of computing and computer professionals. They require some general experience of computing of their readers, but nothing too advanced. Their informal treatment contains plenty of examples so that ideas can be seen in a concrete form. This thoroughly British approach seems to me to make it much easier to understand the subject, being much more down to earth than the approach adopted by the best of the comparable American texts, such as Winston's **Artificial Intelligence**.

The book begins by showing how knowledge processing develops from data processing. Those familiar with 'modern maths' will be at an advantage when reading about sets and relations, but appreciation of these concepts makes the link between the different kinds of processing quite clear. In a computer language, the classification of sets of similar objects is usually achieved by providing a label (or a name) and an index. The label defines the class and the index an instance between objects that have already been classified. In a system for knowledge processing there are classifications and relationships; there are rules and a control structure as well. The classifications and relation ships form the Database. The rules have the form IF (condition) THEN (action). The control structure determines which rule is tried next. When the rule us applied, (condition) is a test of the current state of the database, and (action) updates the database in some way.

But if this shows the underlying similarity, there has been a change in the style of processing. Conventional processing is program-driven with a program always deciding what happens next. Knowledge processing is



data-driven, with the data deciding. Alty and Coombs illustrate the difference between the two styles by comparing the way a real-time system works with the way people work. In a real-time system, an 'interrupt' causes the current situation to be saved and an interrupt routine to be loaded and run. When this routine is completed, the original situation is retrieved and processing carries on from there. People don't work like this. When interrupted, we deal with the interruption, sometimes forgetting what we were doing before. We are data-driven, and this can be a very efficient way of getting things done.

This book proceeds to put more flesh on the bones that were only described in the first one. The way that new knowledge can be generated is dealt with by a treatment of inference based on formal logic (or *predicate* logic). If the thought of a chapter on formal logic seems less than enthralling, do not be discouraged. The treatment is informal, but the reward for perseverance is a neat drawing-together of all the threads of knowledge representation, structure, and deducing new knowledge.

Other ways of representing knowledge, such as frames and semantic networks, are also covered. The explanations are again much more accessible than those in Winston's book referred to earlier, although they are also deliberately less ambitious. The treatment of Prolog towards the end of the book completes a successful explanation of the concepts of expert systems and of the way that they are interlinked.

The second part of the book gives a classification of existing expert systems. I have come across nothing better by way of a technical introduction, and it is suitable for those interested in becoming either users or creators of expert systems.

Expert systems: principles and case studies edited by Richard Forsyth, is directly comparable to the previous book in its aims and content. It is a collection of chapters written by various authors, and this is a source both of strength and of weakness. It suffers from a lack of coherence and from duplication, but it is enlivened by the varying styles and opinions of its contributors.

The potted biographies of the contributors show them to be a singularly distinguished group. Some are academics and most of the remainder were also, (the noted class of ex-Polytechnic lecturers is well represented). Some have started their own companies, including Warm Boot Limited and Green Valley Software, and others indulge in activities as diverse as advising a West German football lottery and writing computer manuals. It is not surprising that their contributions are individual and lively!

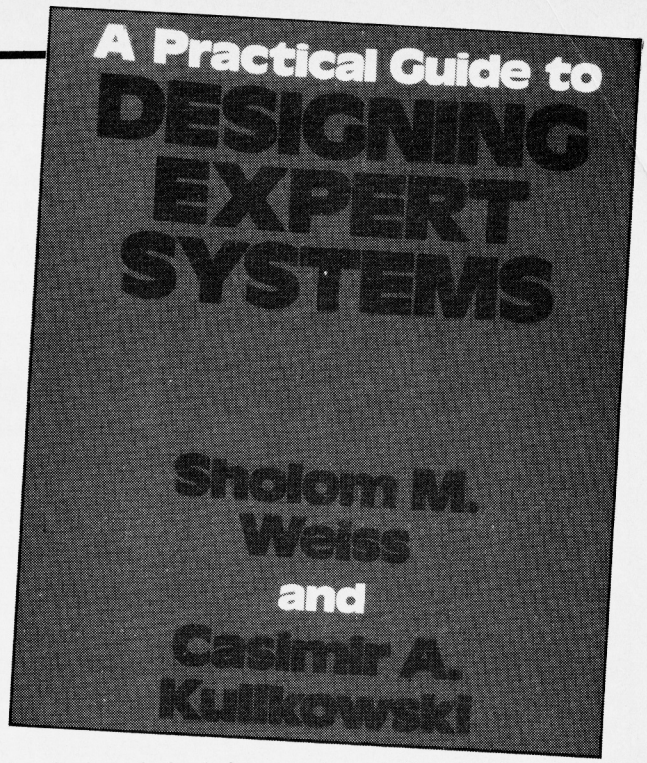
The editor is the most prolific contributor, and his chapters are all very readable. One or two of the others are not, including that by the writer of the computer manual, oddly enough. And one chapter, on fifth generation developments, could have been replaced by something more relevant.

The main area it covers that was not dealt with in the previous book is Machine Learning: ways in which computers can uncover for themselves the rules that apply in a particular domain. They can do this by examining, and then generalising from, instances of knowledge occurring in that domain. Knowledge acquisition is the bottleneck in the development of expert systems, and automating the process is one way of alleviating the congestion. A major problem is to discover the rules in such a form that they can help to provide meaningful explanations for the actions of a system. Apparently — and this is quite extroadinary — one such machine - learning system, on noticing that rules devised by people tended to be better than its own, devised the rule:

> IF a rule is machine-made
> THEN delete it

- I leave you to ponder on this, and to infer what the system did next!

There is a fascinating chapter on 'How shall we judge an expert system?' It takes two approaches to answering the question. The first works from the assumption that expert systems will perform according to expectation, and speculates on the consequences. These can be condensed into the following scenario: If the government has access to knowledge, and proceeds by making logical decisions based on this knowledge, when that knowledge can be stored in a computer and the decisions made by the computer, do we need a government?



The second approach takes the question head on, and quickly shows that it is not at all easy to assess an expert system, or even to compare two of them. The value of an expert system is inextricably linked with its area of application. Expert systems for discovering a scarce and limited resource will be of no value at all when the resource has run out, and their value may come to be seen as entirely negative if they encourage its premature exhaustion. On the other hand expert systems for enduring problems, such as the diagnosis of human illness, are undoubtably worthwhile and their performance can be compared in some sense just as that of human specialists can.

The chapter goes on to demonstrate the dubious nature of inference systems based on fuzzy logic, which rather undermines the two chapters that follow, which are uncritical tutorial accounts of exactly this.

As a text on expert systems, this one lacks the coherence, balance and completeness of that of Alty and Coombs. On the other hand, it is lively to read and it complements the other book rather well. I would buy Alty and Coombs first, but it would be nice to have both.

A practical guide to Expert Systems by S M Weiss and C A Kulikowski covers much the same ground as the previous pair of books, but at a rather higher level and with a higher entry level. This is much more an academic text book than the others. But it is a text on how to write an expert system; it is also appropriately narrow in its coverage, reflecting the authors' preferred approach to their subject. This is much the most advanced of the books. If you have the necessary background and the desire to implement an expert system it is also the book that is most likely to help you.

As a final thought, it appears that in the future we shall consult expert systems with much the same frequency and faith that the ancient Greeks consulted their oracles. In the course of several thousand years and several turns on the spiral of civilisation we still find ways to avoid making decisions.

This month's books are:

Expert systems: A Management Guide by Mike Turner (PA Computers and Telecommunications) 62 pages.

Expert Systems: Concepts and Examples by J M Alty and M J Coombs (NCC Publications) 209 pages.

Expert Systems: Principles and Case Studies edited by Richard Forsyth (Chapman and Hall) 231 pages £9.95

A Practical Guide to Expert Systems by S M Weiss and C A kulikowski, 174 pages, £15.00

INSIDE FASTLOADERS

David Bolton

Fastloaders are (almost) commonplace with commercial games software - and this is how they work...

If you buy Commodore 64 games software, or just look at the adverts and long for Christmas, then you might have noticed a recent development in cassette software - Fast-Loaders.

These appear under a number of different names: Pavloada, Ultra Load, Jetload, Fantasy Loader, Turbo Load, Mega Load and Novaload to name but a few. All do the same job, that is load CBM-64 programs in a much shorter time than the normal Commodore tape routines.

Fast-loaders have been around for several years (since the PET) but it is only in the last nine months that they have emerged in a major way. The main reason for this is that programs are becoming much bigger and more fully exploit the 60K of useable RAM.

It is now the case that most major software houses use fast-loaders and even the duplicators have got into the act and can add fast loading to your programs.

Normal CBM-64 tapes load at about 50 characters per second. A 16K program will load in about six minutes. This compares very badly with ZX Spectrum which will load the same in under two minutes — if we remember that the CBM-64 cassette recorders are all the same and that Spectrum tapes load on a variety of recorders, the situation looks even *worse!*

Under these circumstances fast-loaders have flourished. Most of them can load tapes at 4-10 times normal speed with excellent reliability.

The extra speed has another advantage: it makes tape to tape copying even harder because of the higher frequencies, but duplicators have also had this problem. Some duplication equipment copies at 32 times normal speed and, combined with a fast-loader can give unreliable results.

FACTS AND FIGURES

Before I look at how fast-loaders work there are some relevant and interesting facts about the 64's tape system which are worth mentioning.

Screen Blanking: If you own a '64, you may wonder why the screen is blanked during tape operations. The VIC uses an almost identical system but manages to keep the screen on. This blanking is a result of the VIC II chip used in '64s. It shares the RAM with the 6510 and as this is dynamic RAM it must be constantly refreshed. This is done along with fetching character data during phase 1 of the processor. The 6510 only needs the bus during phase 2, so there is no clash here.

The problem arises when the data is obtained from screen memory. This must be got-at more rapidly than the phase-sharing will allow. To do this the processor must be switched off for a short period, several times per second. This adversely affects the timing, and as a result cassette system will not work reliably; so the answer is to blank the screen.

A SOLUTION

Two fast-loaders that I know have managed to get round this problem. Novaload and Fantasy load can both load with the screen on. This requires extremely fancy programming and Fantasy's solution is based upon two timers synchronising with a cassette pulse. This slows the maximum load speed down to about three times normal speed, but it is still a very good solution. Incidentally, Commodore in the UK are so impressed with Novaload that they are now starting to use it for their own tapes - a real case of selling sand to Arabs.

The VIC manages without screen blanking because it has faster RAM which allows the processor and VIC II chip to co-exist peacefully.

If sprites are turned on they cause the VIC II chip to switch-off the processor for longer periods. I have found it impossible to use the 1541 disk drive with more than three sprites on.

HOW DO FAST-LOADERS WORK?

If you have ever tried to disassemble the Commodore 64's tape routines then you will realise that they are complicated, to say the least. This is to allow for tape speed variations and very extensive error checking.

Each second, roughly 800 bits are sent or received by the tape deck. This works out at 100 cps but Commodore store two copies of a program on tape so effectively reducing it to 50 cps.

Despite the complexity of the tape routine the system is very simple at the hardware end. Think of the tape signal as two levels, low and high. When the signal changes from a high to a low (see fig 1.) then an interrupt is generated.

Upon receipt of an interrupt, the processor reads a hardware clock and obtains the time since the previous interrupt. This time determines if the incoming bit is a '0' or a '1'. This bit is then combined with a string of previous bits and when a byte had been formed, it is stored or compared with memory.

Some fast-loaders use this system but ignore all of Commodore's checking apart from a simple checksum. The timing is also altered for the faster speed and loading at up to 500 cps can be reliably performed.

The Fantasy loader uses a slightly different method, taken from the ZX Spectrum, where the signal is sampled and the number of low or high pulses counted to see if the incoming bit is a '0' or '1'.

A copy of Fantasy's Pyramid for the CBM-64 was partially loaded on a ZX Spectrum. It was able to read most of the tape apart from the header (which is in CBM format). I *didn't* try to run it!

WRITING YOUR OWN FAST-LOADER

If you are reasonably proficient at 6502

machine code then you might have a go at writing a fast-loader.

The only difference between a 6510 which the 64 has, and the 6502, is an 8 bit port at location 1. Bit 4 of this port is connected to the Write head of the cassette deck. If you set this bit then a high signal is written to tape. Clear it and a low signal is sent.

Bit 4 of location 56333 ($DC0D) is connected to the tape Read head and is set when a high to low signal change occurs.

You could set up the interrupt mask at this location and then have an interrupt triggered by a signal change, but this adds extra interrupt handling code. It is easier to sit in a tight loop and monitor the Read bit until it changes.

The main problem I found with the fast loader was not the timing, but getting into synchronisation with the bit-stream being Read. If you start a byte at the wrong bit then your data will be (to put it mildly) incorrect. Solving this is fairly straightforward. Before sending the main data, send a few hundred bytes with a unique bit pattern like 240 (11110000) or 15 (00001111) and Read the bit stream until you get one of these bytes. At that point you are in phase. Now keep on reading these bytes until they change, and that is the first byte of your data.

ERROR CHECKING

How do you ensure that the data you have just Read is the data you sent out? The simplest way is to exclusive OR (EXOR) every byte sent with a checksum byte (which is initially zero) and after the last byte is sent, send the checksum as well.

When loading the tape, every byte is EXORed with another checksum byte, and at the end the tape checksum is EXORed with it. If the result is zero then all is well and you have a better than 90% chance that the tape loaded correctly.

Tape errors can occur but they are usually the result of tape drop-out, dirty or misaligned heads, power fluctuations or mechanical vibration from kicking the cassette deck.

The chances that several errors in combination would cancel out to give the correct checksum are pretty negligible, so if the checksum is correct then you can assume it loaded alright.

FINAL THOUGHT

Many CBM-64 owners have got or are getting disk drives. The 1541 disk drive is a very slow device as all communication is done down a serial port. I have heard of a hardware device which can speed up load and save times and perhaps readers might figure a way to do this in software...

USEFUL ROM ROUTINES

(1) **$F5B3** Prints SEARCHING FOR (name).

Name must be in RAM somewhere pointed to by 187, 188 with the length in 183.

(2) **$F5D2** Prints LOADING or VERIFYING according to whether location 147 is 0 (LOADING) or 1 (VERIFYING).

(3) **$F68F** Prints SAVING (name).

Locations 183, 187 and 188 must be set up as routine.

(4) **$F817** Check if cassette keys are pressed. If not it says PRESS PLAY ON TAPE and waits until you do. It needs no parameters

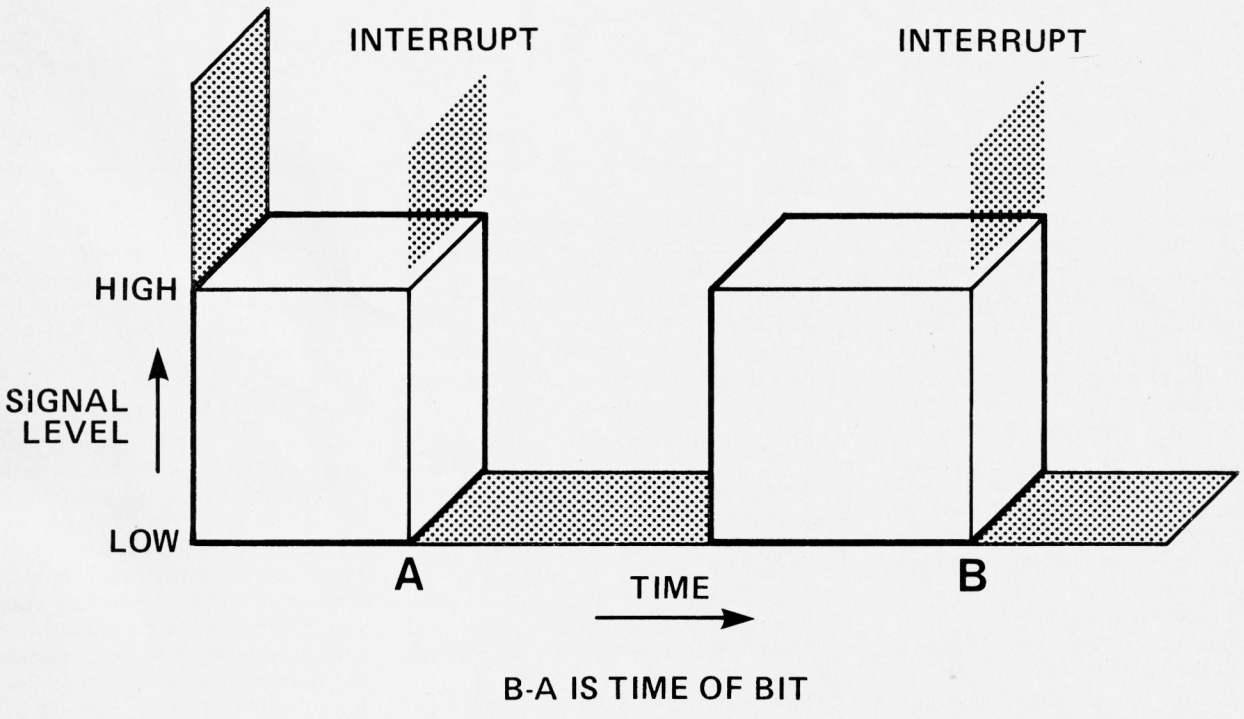
(5) **$F838** Check if cassette keys are pressed. If not it says PLAY AND RECORD and waits until you do.

(6) **$FCCA** Stops cassette motor. This needs no parameters.

There is no separate 'start cassette motor' routine, but the following machine code will do this:

```
;Start   Motor
         LDA 1
         AND £31
         STA 1
;Delay   to give time for motor to speed
         up
         LDX £255
LABEL 1  LDY £255
LABEL 2  DEY
         BNE LABEL 2
         DEX
         BNE LABEL 1
```

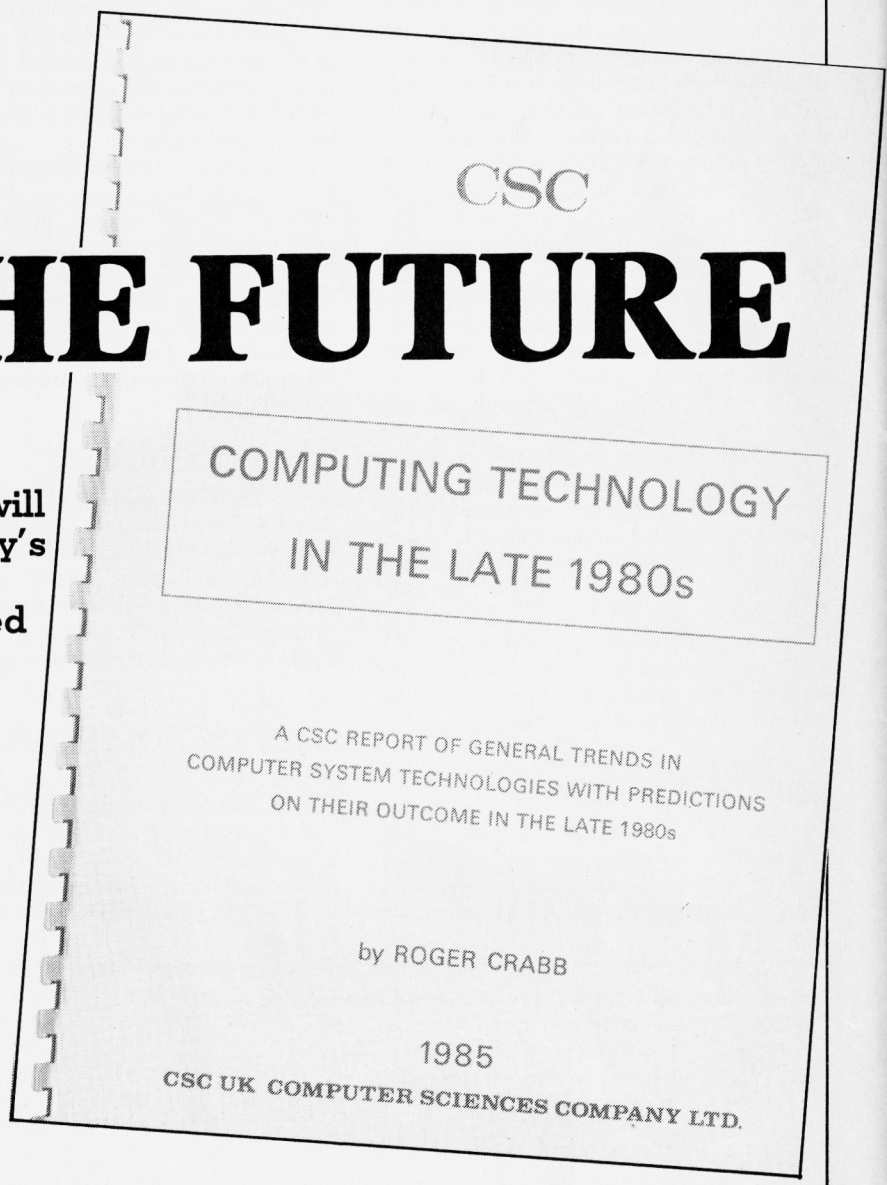
Figure 1: interrupts generated by incoming signal



A LOOK INTO THE FUTURE

Don Thomasson

How computing technology will progress isn't simply 'anybody's guess'. Computer Sciences Company have just published a report predicting, amongst other things, that parallel processors will be standard features by the end of the decade.



In any rapidly-developing field of action, there is naturally an anxious desire to know what the future holds, and this is particularly true of the computer world. *Computing Technology In The Late 1980's*, produced by CSC U.K., seeks to satisfy this desire, and we have been examining its conclusions with interest.

CEC U.K. is the British subsidiary of the Computer Sciences Corporation, and though the report was produced in this country it has something of a transatlantic flavour. However, its strongest praise seems to be reserved for Japanese activities, perhaps reflecting American alarm at the prospect of competition from that quarter.

The reasoning behind the report appears to be roughly as follows;

- In order to honour compatibility commitments, mainstream manufacturers will have to continue to use existing technology as long as possible.
- Meanwhile, hardware design will move ever nearer to known limits of existing technology, allowing and encouraging the creation of very large and fast computers for specialised applications.
- Some of the elements of the extended technology may have a localised impact on mainstream designs.

The first point is entirely valid. In some quarters, it is known as the 'fossilisation syndrome', since it leads to perpetuation of long-obsolete systems on the grounds that any change would involve costly re-programming and the like.

The second point may be valid in itself, but the relevance of highly specialised systems to the computer scene as a whole is perhaps a little uncertain.

The third point is unarguable, but there should be a fourth point, allied to it, suggesting that the fossilised mainstream systems may find themselves competing with physically smaller systems which have used the new technology to enhance their performance. The report pays little attention to microcomputers, as such, seeing the future as the preserve of fairly large computers serving a number of terminal stations. That image persists in some fields of work, but is no longer seen as the only viable approach.

HARDWARE DEVELOPMENT

It is assumed in the report that hardware development will be spurred - on mainly by a desire to increase processing speed and component complexity. These are real enough objects, but other considerations are likely to have an impact on the chosen line of development. Anyone who has watched the battle of the sixteen-bit microprocessors at close range is aware that speed and efficiency have not always been the

main factors in the selection of one type or another. Availability of software has often been the principal criterion, and that has left some of the most interesting devices in the doldrums.

So when 256Kbit RAM components are predicted within two years, to be followed three years later by 1 Megabit RAM, and thereafter by 4 Megabit devices, one can marvel respectfully at such achievements, yet doubt whether economic considerations will encourage widespread sales.

Looking at chip complexity in more general terms, the report envisages integrated circuits with 100 - 300 external connections. This would present serious problems for printed circuit designers, who currently need to limit connection density to about twenty lines per linear inch of width. It would be absurd to have a tiny chip surrounded by a fifteen-inch connection perimeter, so some new approach would be needed. This problem has been increasingly evident

SOFTWARE

The section of the report dealing with software tends to be rather less definitive than the hardware section, but it may well arouse stronger emotions. For example, the statement that 'European and U.S. software is typically less reliable than Japanese during the past five years, and a number of solutions have been tried and discarded because they created other problems. unless a breakthrough can be made in this area, the practical viability of such complex components must remain doubtful. In any case, it is reasonable to question the advantages of such high integration levels. The report touches on the difficulties of locating and correcting design or manufacturing faults, but offers no solutions. System changes become much more difficult and time-consuming with high integration levels, and a world in which system requirements are right first time can only be envisaged by those who have never participated in systems work...

Close association with enthusiasts for very large scale integration has raised a suspicion that they envisage an ideal world in which performance requirements never change (except to suit the needs of the chip designer), errors are never made, and achievement is judged solely on the number of elements packed into a given area. If this sounds cynical, think of a certain component that went through a number of corrective iterations, each taking about six months, and was finally made valid by changing the performance specification. Then there is a very interesting chip that has hovered on the edge of availability for years, continually delayed by a shortage of suitable design staff.

The report ends its consideration of chip technology by predicting self-diagnostic facilities, which would bring spare circuits into action in the event of a failure.

BULK STORAGE

For most of the computer world, bulk (back up) storage implies discs of some sort. The report identifies access time and transfer rate as the main limitations in this area, and no one would argue with that, given the ability of associated hardware to cope with higher rates. A suggested speed of 15-30 megabytes per second may be desirable from a system point of view, but could involve serious hardware problems.

Some interesting points regarding modified magnetic discs and optical discs are made, but it is doubtful whether such techniques will threaten orthodox discs within the period reviewed. Bubble memory is put firmly in its place as mainly suitable for special applications, such as in portable 'battlefield' systems, and there is reference to 'database engines', which simplify search processes.

DISPLAY AND HARD COPY

Where display techniques are concerned, the report concentrates on terminal stations, and ignores some extant devices. Apricot will not be amused by the statement that 'flat panel' terminals of 'commercial quality' may be expected in 1985/6, or the statement that 'In the long term, voice input capability will be introduced'. Perhaps a visit to the P.C.W. Show might have been useful...

The predictions for future displays are not particularly optimistic. Colour displays with 2048 x 2048 resolution are expected to take five years to mature, whereas these figures need to be doubled for some applications. It could be added that current hardware does not always live up to its specified capability, and performance figures in this area are better than the actual results.

Colour printers are mentioned, bringing a memory of a senior manager who banned the use of colour in reports because his division had no access to colour-copying facilities. Such pragmatism has a more effective influence than a whole mass of technological data.

software' may evoke a dusty answer from some British software houses.

Much of the rest of the section reflects the universal struggle to find new and more effective ways of locating and correcting program bugs. There are signs of an uneasy distrust in inadequately-proven methodologies, which will be shared by anyone who has been subjected to an abstruse software theories on the subject.

This is only one area of unease. The paradox of American Department of Defence pressure to use Ada, at a time when there is a shortage of proven compilers for the language is mentioned, and a similar situation in this country could have been cited. Partly because of such situations, an impression is conveyed that the degree of uncertainty is far greater in the software area than with hardware, but experienced hardware engineers may doubt whether that is representative of the real facts of life, feeling that the statements regarding hardware are a little optimistic.

EXPERT SYSTEMS

The report ends with an interesting study of expert systems, beginning with a praiseworthy attempt to define such a system. There is refence to 'Fuzzy Logic', illustrated by examples, and a consideration of PROLOG.

The fascinating thing from our point of view is that a list of problem areas in relation to expert systems agrees very closely with the points put forward by Bill Home in his article on 'The Intelligent Computer'. For once, Bill seems to have found support...

CONCLUSION

Given the assumptions made, the conclusions reached by the report are eminently reasonable, though it might be felt that additional factors could usefully have been taken into consideration. The report has little relevance to the microcomputer field, which may ultimately have a very relevant influence on the computer world as a whole.

AVAILABILITY

Computing Technology In The Late 1980's is available from Publications Department, CSC U.K. Computer Sciences Company Ltd., 20 Savile Row, LONDON W1X 1AE. The price is £95, reduced to £65 for educational establishments.

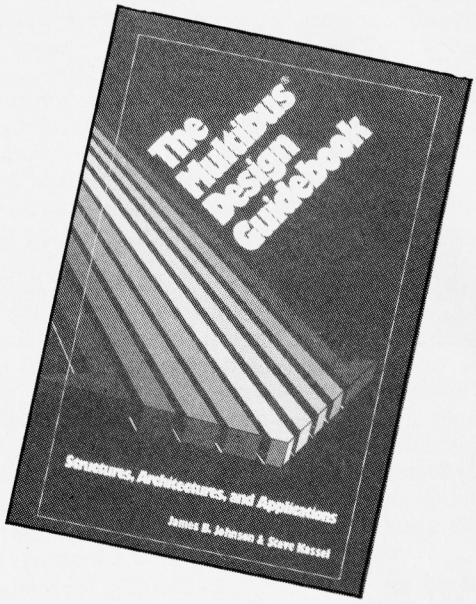
FACTSHEET: Computing Technology In The Late 1980's

Price:
£95 reduced to £65 for educational establishments

Available from:
Publications Department
CSC U.K. Computer Sciences Company Ltd.
20 Savile Row London W1X 1AE

MULTIBUS

Bill Horne



When we first saw *The Multibus Design Guidebook*, our immediate reaction was that it would be of little interest to our readers, but we changed our minds. The reason for the change will appear in due course — but, for a start, what is MULTIBUS?

Building a specialised computer system can be considerably simplified if it is made up from standard ready-made boards, but there is a snag. The boards must be compatible with one another.

One way of ensuring compatibility is to use boards which conform to a well-defined interface standard, such as S50, S100, Nasbus - or MULTIBUS, which is perhaps the most comprehensive of them all. It defines the electrical connections, the timing protocols, and even the shape and size of the printed circuit boards. This allows quite large systems to be created with the minimum of design work, but the very completeness of the specification can be a disadvantage with smaller systems. To see the reason for this, we need to go back to the origins of the MULTIBUS concept.

ORIGINS

During the 'seventies, the Intel Corporation were much involved in early microprocessor systems, starting with the little 4004. In those days, everyone wanted to build their own systems, but for that they needed development systems, and these they could buy from Intel. To meet the need for variants to suit particular customers, Intel required a standard, and the original MULTIBUS was introduced in the Intellec 800, around 1975. By building to the given specification, users could add peripheral interfaces and other specialities with confidence that they would work - if they met the specification.

The date of introduction is important, because in 1975 the process of increasing MULTIBUS board were influenced by the integration scale in microchips had not really got under way. In ten years, the number of transistors in a chip rose from 2000 to 100,000, rising by some 50% each year. In 1975, the figure was around 5000. integration scale had a strong influence on chip count, and that, in turn, affected printed circuit board size. The dimension of the situation in 1975, which suggested that 12" x 6.75" was appropriate.

FACTSHEET:
The MULTIBUS Design Guidebook

Authors:
James B Johnson and Steven Kassel

Publisher:
McGraw Hill Inc.

Price:
£26.75

Hardback

At the time, there was a great deal of acrimonious discussion about printed board design, because there were two opposing sets of requirements. From the mechanical point of view, it was desirable to support a board by its longer edges to minimise the unsupported span. System designers, on the other hand, wanted to use the longer edges for connections, because that gave them more space to play with and reduced the average length of internal tracks.

For situations where rigorous environmental conditions reinforced the mechanical point of view, it was possible to arrange a compromise, the board being made up into a highly rigid box-like structure, with external connections on lone edges. This, however, was an expensive approach. Since Intel did not envisage the use of MULTIBUS in severe environments, they chose to put the connectors on the longer edge of the boards, using double-sided board-edge connectors. The P1 connector provided 86 lines, the P2 connector 60 lines.

In present-day practice, with increased chip complexity, reduced chip count, and consequently more compact systems, both the board size and the number of connections may seem an overkill to those who are used to more compact systems, but MULTIBUS was aimed at a higher target.

THE INTERFACE

The main interface lines use the P1 connector. Power supplies at +5v,+ and —12v are catered for, with multiple pins for each voltage level. Signal Ground and 0V are considered common, which can be an embarrassment with devices that draw high peak currents, but such devices are rare these days.

Twenty address lines are provided, with another four as an optional extension on the

P2 connector. There are sixteen data lines. For interrupt there are eight demand lines and one acknowledge line. The rest of the lines are used for qualifiers and controls. These allow ROM and RAM to be enabled or disabled independently, extending the effective addressing range, and provide for transfer of bus control to a temporary master. This allows DMA (Direct Memory Access).

To keep the interface lines simple, the boards are normally mounted side by side, the lines running from connector to connector without crossing over. The boards are slid into a frame, which is commonly of cast metal, and the overall arrangement is pleasantly neat, if sometimes a little bulky.

For testing and fault-finding it is necessary to use 'extender' boards, which replace the normal boards and allow the latter to be plugged into the outer end of the extender, where they are more accessible. This usually presents no problems, especially if the extender tracks are formed as 'striplines', being placed near an earthed plane and proportioned to give the optimum impedence.

Power supplies are mounted separately, and keyboards and displays are served as peripherals. Like other peripheral devices, they connect to the 'outer' end of the standard boards containing the interface which serves them.

VARIANTS

With so comprehensive a set of basic lines, it might seem that all possible needs are covered, but Intel took a different view.

They noted that data flow could be a problem, especially where high-speed input/output was involved, so they created the Multichannel Bus. This was completely independant from the main MULTIBUS, and was used only for input/output data, transferring up to 8 megabytes per second over lines up to fifteen metres in length. The P2 connector is used, and there are sixteen lines carrying multiplexed addresses and data, plus controls and qualifiers. The system allows input/output data to be transferred independently from the main MULTIBUS, which can be handling other work.

The second system extension allowed input/output interfaces to be plugged in, 'piggyback' fashion, to the main boards allowing extensions to be added without the need to add another full sized board. This is the iSBX interface.

Finally, the iLBX interface caters for a situation in a multi-processor system where a given processor has its own local store in addition to accessing a global store shared by all other processors. This interface covers the local link.

THE GUIDEBOOK

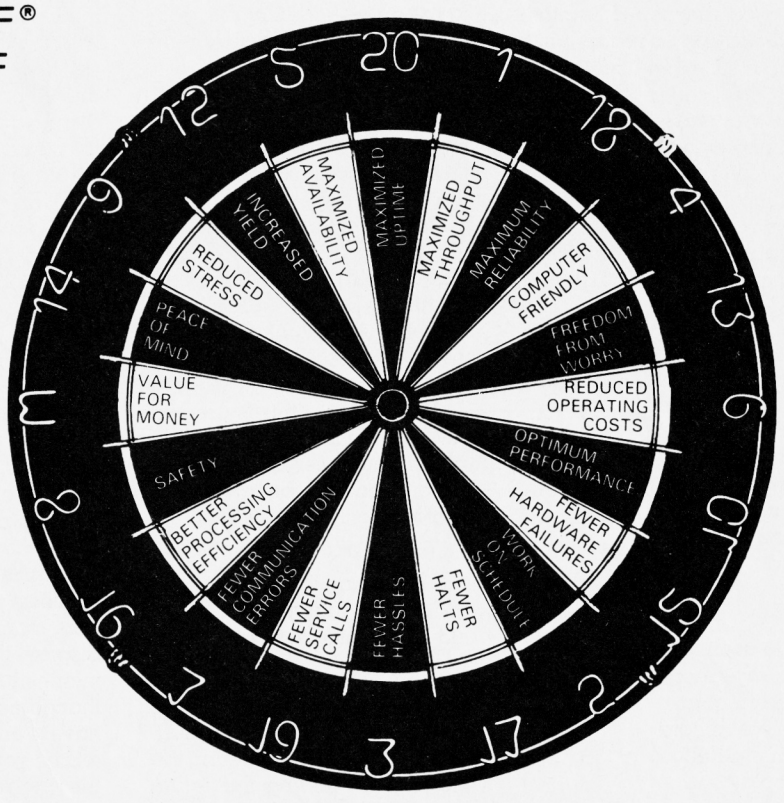
The above notes, compounded of shavings from the Guidebook and personal experience of MULTIBUS systems, will probably have painted a picture that has little appeal to even the more enthusiastic constructors among our readers, but that does not mean that the Guidebook is of no interest. It begins by studying the general concept of MULTIBUS, and goes on to examine the detailed requirements. Midway through the 400 pages, however, a more general study begins, starting with a chapter on single-board computers, continuing with chapters on multiprocessing and multicomputer systems, and ends with chapters on system and board design. Since the book was written by two members of Intel's Microcomputer Systems Division, it is scarcely surprising that these chapters have a slant towards Intel ideas, but that does not detract from the value of the information provided - information which it would be difficult to find in such compact form elsewhere.

For anyone who has occasion to work with MULTIBUS, the book is essential. The only doubt here is that each section on MULTIBUS and its variants begins by urging the wisdom of obtaining the latest specifications, which suggests a lack of stability that has not been found in practice. There have been changes, of course, but nothing too drastic.

For anyone who wants to know how to organise multiprocessing, or put more than one processor into a system, or set up cache memory, or simply find out how to put a system together, the book is also useful. It is by no means cheap, but one could waste as much money in no time by buying something less costly.

A book mainly intended for the professional, then, but one which holds semi-professional and amateur appreciatable interest for the

WHICH ONEAC® MAINS CONDITIONING BENEFITS SCORE HIGHEST WITH YOU?



CONTACT YOUR ONEAC POWER PROFESSIONAL AT:

ONEAC® LIMITED
Unit 6, Eyston Way,
Abingdon, Oxon, OX14 1TR

Abingdon (0235) 34721
Telex: 837976 ONEAC G.

CLUBFILE

R A. Lines and
R. Lines

A filing system/database for Atari micros.

Clubfile is a small filing system/database which allows you to keep track of names and addresses. It was developed after the number of members of a local youth club we help to run, started to approach forty, which became a bit of a headache using the old, manual, system.

When you first RUN the program, you are asked whether you wish to Load a file or Start a new file. A new user would select the Start a file option which would take the user to the Add Record section.

After the original start in the Load section, the program will always return to the two main menus, these being the search menu and the file menu:

Search menu
Change menu
Age search
Name search
Save file
Load file
Zone search
End

File menu
Change menu
Add Record
Delete Record
Amend Record
Display
End

The required option is chosen by moving a joystick in port number one. As you move it up and down the available option is highlighted. When the desired option is highlighted, the joystick trigger should be pressed, thereby selecting the option.

The following are brief explanations of each option available:

1. **Change menu** — this simply transfers you from one menu to the other. For example, from the search menu to the file menu and vice versa.

2. **Add Record** — this adds a record to the file. Please note that when you use clubfile for the first time, this is the option that enters M for males and F for females, this will lead to Mr. and Miss being printed. For additional titles just add-in additional IF statements in the PRE$ setups (eg. Lines 970-990).

3. **Save file** — this is used to save the file in memory to disk. When you enter the Save file option you will be given the choice of:

1 Returning to the menu
2 Exiting the program
3 Continuing to save the file

Selecting to continue to save the file will lead to the production of a prompt for entering the file name. Pressing the return key will save the file under the default file name displayed. To enter a different file name, just edit the file name displayed and press the RETURN key.

4. **Load file** — this is used to load the file from disk storage into memory for use. When you enter the load file section you will be asked whether you wish to Load a file or Start a new file. If you selected the Load file option then the program will check the disk for the names of the data files on it and print them out on the screen. Then a prompt for the file name will appear along with the default file name. Pressing the RETURN key will load the file under the default name displayed. To enter a different file name, just edit the file name displayed then press the RETURN key.

5. **Zone search** — this allows you to find all of the members in a particular zone. Please note that this is a **printer only** function.

6. **Age search** — this option requests you to input the date and then all members who are either too old or too young for the club are displayed.

7. **Name search** — this option requests you to input the members name, and then either displays the members details or informs you that no member of that name is on file. You do not have to enter the full name of the record you are searching for — the program will look for all names beginning with whatever input you enter. For example if you enter the letter P then it will display all names beginning with P.

8. **Display** — when you select the display option, you will be given three choices of display:

a) Screen Option(1), will display the records in alphabetical order on the screen.
b) Printer Option(2), prints out an attendance sheet.
c) Printer Option(3) prints out the details in a list form.

9) **Delete Record** — this deletes any members record from the file. To select the record, use a search option to find member, then leave search option, and select delete. The record will be displayed and you will be asked if you wish to delete the file; type Y to do so.

10. **Amend record** — this amends any members records already on file. Similar to delete for selection of file.

11. **End** — this returns you to BASIC after reminding you to save the file to disk.

Cassette users will need to make a few changes in the load and save sections, these are as follows:

Line 200 — Remove ,FILE$(15)
Line 240 — Remove :FILE$="D:CLUBFILE.DAT"
Line 2670 — change the word diskette to cassette
Lines 2720-2740 should be deleted.
Line 2705 — FILE$ should be replaced by "C:"
Lines 2780-2810 should be deleted.
Line 2830 — GOTO 2860 is placed by GOTO 2880.
Lines 2860 and 2870 should be deleted.
Line 2880 — FILE$ should be replaced by "C:"

The printer options of the program have been written to work with an Epson RX-80 Printer. The codes required may well be different on different printers, for this reason we have listed the printer codes used and the functions they perform:

CHR$(27):"m": CHR$(4) — GRAPHICS MODE. If your printer does not have a graphics mode replace the graphics characters with an ordinary character, for example, an exclamation mark etc

CHR$(27):"E" — This sets

the printer in its highest quality print mode.

CHR$(27):"8" — This sets the paper end ignore feature. Paper end ignore would stop the printer when the paper is close to running out.

CHR$(27):"1" — This is the line space control, it sets the spacing so that there is no space between characters. CHR$(27):"2" — sets line spacing back to normal.

CHR$(27):"—":"n" — n=0 stops underline mode.

LINE 240 contains the graphic characters to be printed for grid pattern on attendance sheet (the only thing that requires graphic mode). The first character is the left half of a square, the second is a complete square. The characters typed in the program look different to this and are obtained by typing, Inverse Control M, and Inverse Control L.

Due to the Atari's unique string manipulation techniques (it makes use of strings of up to 32K rather than string arrays), it would prove to be almost impossible to convert this program for use on other computers.

The program as shown will only handle fifty records but this may be extended by altering the DIM statement in line 200 of WFILE$ by any multiple of 93, up to the limit of available memory. To change the age limits the values for YOUNG and OLD in line 2340 the equations Z—(Age—1)*10000 for YOUNG and Z—(Age+1)*10000 for OLD should be used.

In the program listing shown, all the inverse characters are represented by italics.

Variable List

A$	Temporary storage and input.
AD1$,AD2$	Entry of address.
AD3$	Entry of address and input of Directory.
ADD$	Holds Address of individual record.
B$,I$,T$	Temporary storage.
DOB$	Holds Date of Birth of individual record
FILE$	Holds Filename
INIT$	Holds Initial of individual record.
MULTI$	General input string.
PFILL$, PFORM$	Holds Printer output.
POUT$	Used for formatting output.
PRE$	Used for setting title.
S1$, S2$, S3$, S4$	Used for messages in menu.
S5$, S6$, SB2$, SB3$	Used for messages in menu
SB4$, SB7$, SB8$,	Used for messages in menu.
SB9$, SU0$, SE1$	Used for messages in menu
SU2$, SU3$, SU4$	Used for messages in menu.
SU5$, SU6$, SU7$,	Used for messages in menu.
SU8$, SU9$	Used for messages in menu.
SEARCH$	Used for comparison with SURN$.
SURN$	Holds Sumame of individual record.
TELE$	Holds Telephone number for individual record.
WFILE$	Single string holds complete file.
ZONE$	Holds zone code for individual record.
C, I	Used for determining positions in records.
CTR	Counter and in FOR/NEXT loops.
D	Counter and PRINT positioning.
DEL, p, Q, V, X, Y	FOR/NEXT loops.
I1	Second variable for second record.
J	Size of WFILE$
JY	Y co-ordinate of cursor in menus.
XP	X co-ordinate of cursor in menus.
K	Flag set in delete for use in sort.
M, R	Used for input.
OLD	Used for age limit.
YOUNG	Used for age limit.
S	Counter
ST	Holds value from STICK(0) command.
TEST	Flag in sort routine.
TRIG	Holds value from TRIG(0) command.
TY	Holds value for JY, used to check if stick has been moved since last pass.
VV	Holds length of strings.
Z	Holds value of date.

Listing: Clubfile

```
100 REM SOME OF THESE LINES ARE TOO        LONG TO ENTER ON DEFAULT SCREEN SO
  REMEMBER TO POKE 82,0
110 OPEN #6,12,0,"S:":SETCOLOR 2,14,0:POKE 82,2:GOTO 200
120 D=0:FOR P=1 TO 60:POUT$=ADD$(P,P):IF POUT$="," THEN 140
130 ? POUT$;:GOTO 150
140 D=D+1:? POUT$:POSITION 7,9+D
150 NEXT P
160 RETURN
170 REM REPEAT CHOICE SUB-ROUTINE
180 POSITION 6,20:? "Press SPACE BAR to CONTINUE";
190 POSITION 4,22:? "Press ANY KEY to return to menu";:OPEN #2,4,0,"K:":GET #2,R
:CLOSE #2:RETURN
200 DIM WFILE$(4650),SURN$(16),INIT$(2),ADD$(60),DOB$(10),TELE$(8),A$(16),B$(16)
,T$(93),ZONE$(1),FILE$(15):J=0
210 DIM SB2$(10),SB3$(13),SB4$(12),SB7$(10),SB8$(13),SB9$(12),MULTI$(100),AD1$(7
0),AD2$(20),AD3$(20),I$(93),S1$(3)
220 DIM POUT$(2),SEARCH$(16),S2$(3),S3$(11),S4$(11),S5$(7),S6$(7),PFORM$(40),PFI
```

```
LL$(30),PRE$(5)
230 S1$="End":S2$="End":S3$="Zone search":S4$="Zone search":S5$="Display":S6$="D
isplay"
240 PFORM$="▮  ▮  ▮  ▮  ▮  ▮  ▮  ▮  ▮  ▮  ▮  ▮  ▮":FILE$="D:CLUBFILE.DAT"
250 PFILL$="........................."
260 SB2$="Add record":SB3$="Delete record":SB4$="Amend record":SB7$="Add record"
:SB8$="Delete record":SB9$="Amend record"
270 DIM SU1$(11),SU2$(10),SU3$(11),SU4$(9),SU5$(9),SU6$(11),SU7$(10),SU8$(11),SU
9$(9),SU0$(9)
280 SU6$="Change menu":SU7$="Age search":SU8$="Name search":SU9$="Save file":SU0
$="Load file"
290 SU1$="Change menu":SU2$="Age search":SU3$="Name search":SU4$="Save file":SU5
$="Load file":GOTO 2770
300 REM SORT ROUTINE
310 TRAP 360:POSITION 14,23:? "Working.../";:TEST=0
320 FOR C=1 TO J/93-1:I1=C*93:A$=WFILE$(I1-92,I1-76):B$=WFILE$(I1+1,I1+16):IF A$
<=B$ THEN 340
330 TEST=TEST+1:T$=WFILE$(I1-92,I1):WFILE$(I1-92,I1)=WFILE$(I1+1,I1+93):WFILE$(I
1+1,I1+93)=T$
340 NEXT C:IF TEST<>0 THEN 310
350 IF K=1 THEN J=J-93
360 ? CHR$(253):K=0:RETURN
370 REM DELETE RECORD ROUTINE
380 C=I:GOSUB 2120:POSITION 13,1:? "Delete Record":POSITION 3,20:? "Do you wish
to delete this Record?";
390 OPEN #1,4,0,"K:":GET #1,R:CLOSE #1:IF R<>ASC("Y") THEN 1410
400 FOR K=C TO C+92:WFILE$(K,K)="Z":NEXT K:K=1:GOSUB 310:GOTO 1410
410 REM AMENDMENT ROUTINE
420 TRAP 2100:C=I:PRINT CHR$(125)
430 POSITION 14,1:? "Amend Record":POSITION 10,20:? "Enter correct lines";:POSIT
ION 2,22:? "Change lines using edit"
440 POSITION 0,4:? "Surname: ";:POSITION 0,5:? "Initial: ";:POSITION 0,6:? "Date
 of birth (DDMMYY): ";
450 POSITION 0,9:? "Road: ";:POSITION 0,7:? "Phone number: ";:POSITION 0,8:? "Zo
ne: ";:POSITION 0,10:? "Area: ";
460 POSITION 0,11:? "Town: ";:SURN$=WFILE$(C,C+15):POSITION 10,4:? SURN$;:INIT$=
WFILE$(C+16,C+16):POSITION 10,5
470 ? INIT$
480 DOB$=WFILE$(C+18,C+23):POSITION 25,6:? DOB$;:ADD$=WFILE$(C+24,C+83)
490 TELE$=WFILE$(C+84,C+91):POSITION 15,7:? TELE$;:ZONE$=WFILE$(C+92,C+92):POSIT
ION 7,8:? ZONE$:POSITION 7,9:GOSUB 120
500 GOSUB 1670:WFILE$(C,C+15)=SURN$:WFILE$(C+16,C+17)=INIT$:WFILE$(C+18,C+23)=DO
B$:WFILE$(C+24,C+83)=ADD$
510 WFILE$(C+84,C+91)=TELE$:WFILE$(C+92,C+92)=ZONE$:GOSUB 310:GOTO 1410
520 REM ZONE SEARCH ROUTINE
530 PRINT CHR$(125):POSITION 14,1:? "Zone search";:POSITION 8,6:? "Which zone co
de (A/B/C) ";:INPUT MULTI$
540 IF LEN(MULTI$)<>1 THEN 530
550 SEARCH$=MULTI$
560 OPEN #2,8,0,"P:":? #2;CHR$(27);"E";CHR$(27);"8"
570 ? #2;CHR$(27);"-";"1";"ZONE :";SEARCH$;CHR$(27);"-";"0":CLOSE #2
580 FOR I=1 TO J STEP 93:IF SEARCH$<>WFILE$(I+92,I+92) THEN 610
590 INIT$=WFILE$(I+16,I+17):SURN$=WFILE$(I+1,I+15):ADD$=WFILE$(I+24,I+83)
600 OPEN #2,8,0,"P:":? #2;INIT$;SURN$;ADD$:CLOSE #2
610 NEXT I
620 REM SEARCH SUB MENU
630 ? CHR$(125):XP=13:JY=7:ST=0
640 POSITION 14,1:? "Search menu":GOTO 700:GOSUB 790
650 ST=STICK(0):JY=JY+(ST=13)+(ST=9)+(ST=5)-(ST=6)-(ST=10)-(ST=14):TRIG=STRIG(0)
660 IF JY<7 THEN JY=7
670 IF JY>13 THEN JY=13
680 IF TRIG=0 THEN 820
690 IF TY=JY THEN 650
700 POKE 755,0:GOSUB 790:POSITION XP,JY
710 IF JY=7 THEN ? SU6$;
720 IF JY=8 THEN ? SU7$;
730 IF JY=9 THEN ? SU8$;
740 IF JY=10 THEN ? SU9$;
750 IF JY=11 THEN ? SU0$;
```

```
760 IF JY=12 THEN ? S4$;
770 IF JY=13 THEN ? S2$;
780 POKE 755,2:TY=JY:FIRST=0:GOTO 650
790 POSITION XP,8:? SU2$;:POSITION XP,9:? SU3$;:POSITION XP,10:? SU4$;
800 POSITION XP,11:? SU5$;:POSITION XP,13:? S1$;
810 POSITION XP,12:? S3$;:POSITION XP,7:? SU1$;:RETURN
820 REM BRANCH FROM MENU
830 ON JY-6 GOTO 1410,2260,2480,2660,2770,520,2660
840 GOTO 650
850 REM DISPLAY ROUTINE
860 ? CHR$(125):POSITION 13,1:? "DISPLAY FILES":POSITION 13,5:? "SCREEN OPTION (
1)";
870 POSITION 13,6:? "PRINTER OPTION (2)";
880 POSITION 13,7:? "PRINTER OPTION (3)";
890 POSITION 5,9:? "CHOOSE OPTION ";:INPUT M
900 IF M=1 THEN 1360
910 IF M=3 THEN 1170
920 IF M<>2 THEN 860
930 OPEN #2,8,0,"P:":? #2;CHR$(27);"m";CHR$(4);CHR$(27);"E";CHR$(27);"8";CHR$(27
);"1":GOSUB 940:GOTO 960
940 ? #2;"DATE                          ";PFORM$:? #2;"                                ";PFO
RM$
950 ? #2;PFILL$;CHR$(27);"-";"1";PFORM$;CHR$(27);"-";"0":CTR=0:RETURN
960 FOR I=1 TO J STEP 93:INIT$=WFILE$(I+16,I+17):SURN$=WFILE$(I+1,I+15):CTR=CTR+
1
970 IF WFILE$(I,I)="M" THEN PRE$="Mr.  ":GOTO 1000
980 IF WFILE$(I,I)="F" THEN PRE$="Miss.":GOTO 1000
990 PRE$="      "
1000 ? #2;"                                 ";PFORM$:? #2;PRE$;INIT$;SURN$;"    ";PFORM$
1010 ? #2;PFILL$;CHR$(27);"-";"1";PFORM$;CHR$(27);"-";"0"
1020 IF CTR<>20 THEN 1070
1030 GOSUB 1110:? CHR$(253);CHR$(125):POSITION 6,6:? "POSITION NEW SHEET OF PAPE
R":POSITION 6,7:? "AND PRESS RETURN"
1040 CTR=0:POKE 764,255:OPEN #3,4,0,"K:":GET #3,R:CLOSE #3:IF R=155 THEN 1060
1050 GOTO 1040
1060 GOSUB 940
1070 NEXT I
1080 FOR I=1 TO 5:? #2;"                                    ";PFORM$
1090 ? #2;"                                 ";PFORM$
1100 ? #2;PFILL$;CHR$(27);"-";"1";PFORM$;CHR$(27);"-";"0":NEXT I:GOSUB 1110:GOTO
 1160
1110 ? #2;"                                 ";PFORM$
1120 ? #2;"SUBS TOTAL                       ";PFORM$
1130 ? #2;"                                 ";PFORM$
1140 ? #2;"                                 ";PFORM$
1150 ? #2;PFILL$;CHR$(27);"-";"1";PFORM$;CHR$(27);"-";"0":RETURN
1160 ? #2;CHR$(27);"2":CLOSE #2:GOTO 1410
1170 OPEN #2,8,0,"P:":? #2;CHR$(27);"E";CHR$(27);"8";CHR$(27);"1":CTR=0
1180 FOR I=1 TO J STEP 93
1190 CTR=CTR+1
1200 FOR V=1 TO 60
1210 IF WFILE$(I+V+24,I+V+24)="," THEN GOTO 1230
1220 NEXT V
1230 ADD$=WFILE$(I+24,I+V+24):INIT$=WFILE$(I+16,I+17):SURN$=WFILE$(I+1,I+15):TEL
E$=WFILE$(I+84,I+91)
1240 DOB$(1,2)=WFILE$(I+18,I+19):DOB$(3,3)="/":DOB$(4,5)=WFILE$(I+20,I+21):DOB$(
6,6)="/":DOB$(7,8)=WFILE$(I+22,I+23)
1250 V=LEN(ADD$):FOR VV=V TO 24
1260 ADD$(VV+1,VV+1)=" ":NEXT VV
1270 IF WFILE$(I,I)="M" THEN PRE$="Mr.  ":GOTO 1300
1280 IF WFILE$(I,I)="F" THEN PRE$="Miss.":GOTO 1300
1290 PRE$="      "
1300 ? #2;PRE$;INIT$;SURN$;"  ";ADD$;TELE$;"    ";DOB$:? #2;CHR$(10)
1310 IF CTR<>20 THEN 1350
1320 ? CHR$(253);CHR$(125):POSITION 6,6:? "POSITION NEW SHEET OF PAPER":POSITION
 6,7:? "AND PRESS RETURN"
1330 CTR=0:POKE 764,255:OPEN #3,4,0,"K:":GET #3,R:CLOSE #3:IF R=155 THEN 1350
1340 GOTO 1330
1350 NEXT I:? #2;CHR$(27);"2":CLOSE #2:GOTO 1410
```

```
1360 PRINT CHR$(125):FOR I=1 TO J STEP 93
1370 GOSUB 2120:POSITION 13,1:? "DISPLAY FILES":GOSUB 180
1380 IF R<>ASC(" ") THEN 1410
1390 NEXT I
1400 REM FILE SUB MENU
1410 ? CHR$(125):XP=15:JY=7
1420 POSITION 16,1:? "File menu":GOTO 1480:GOSUB 1560
1430 ST=STICK(0):JY=JY+(ST=13)+(ST=9)+(ST=5)-(ST=6)-(ST=10)-(ST=14):TRIG=STRIG(0
)
1440 IF JY<7 THEN JY=7
1450 IF JY>12 THEN JY=12
1460 IF TRIG=0 THEN 1580
1470 IF TY=JY THEN 1430
1480 POKE 755,0:GOSUB 1560:POSITION XP,JY
1490 IF JY=7 THEN ? SU6$;
1500 IF JY=8 THEN ? SB7$;
1510 IF JY=9 THEN ? SB8$;
1520 IF JY=10 THEN ? SB9$;
1530 IF JY=11 THEN ? S6$;
1540 IF JY=12 THEN ? S2$;
1550 POKE 755,2:TY=JY:GOTO 1430
1560 POSITION XP,8:? SB2$;:POSITION XP,9:? SB3$;:POSITION XP,10:? SB4$;
1570 POSITION XP,11:? S5$;:POSITION XP,12:? S1$;:POSITION XP,7:? SU1$;:RETURN
1580 REM BRANCH FROM MENU
1590 ON JY-6 GOTO 630,1610,370,420,850,2660
1600 GOTO 1430
1610 REM ADD RECORD
1620 IF J=4650 THEN ? CHR$(253):GOTO 1410
1630 PRINT CHR$(125):POSITION 16,1:? "Add Record";
1640 POSITION 0,4:? "Surname: ";:POSITION 0,5:? "Initial: ";:POSITION 0,6:? "Dat
e of birth (DDMMYY): ";
1650 POSITION 0,9:? "Road: ";:POSITION 0,10:? "Area: ";:POSITION 0,11:? "Town: "
;
1660 POSITION 0,7:? "Phone number: ";:POSITION 0,8:? "Zone: ";:GOSUB 1670:GOTO 2
050
1670 POSITION 9,4:INPUT MULTI$:IF LEN(MULTI$)=0 THEN 1670
1680 IF LEN(MULTI$)<17 THEN 1710
1690 POSITION 13,14:? "Name too long";:? CHR$(253):FOR 0=1 TO 20:NEXT 0:POSITION
 13,14:? "                "
1700 POSITION 9,4:? "                                        ";:GOTO 1670
1710 IF LEN(MULTI$)<16 THEN MULTI$(LEN(MULTI$)+1)=" ":GOTO 1710
1720 SURN$=MULTI$
1730 POSITION 9,5:INPUT MULTI$:IF LEN(MULTI$)=0 THEN 1730
1740 IF LEN(MULTI$)<2 THEN 1780
1750 POSITION 12,14:? "One initial only";
1760 ? CHR$(253):FOR 0=1 TO 20:NEXT 0:POSITION 12,14:? "                    ";:POSIT
ION 9,5:? "       ";
1770 GOTO 1730
1780 IF LEN(MULTI$)<2 THEN MULTI$(LEN(MULTI$)+1)=".":GOTO 1780
1790 INIT$=MULTI$
1800 POSITION 24,6:INPUT MULTI$:IF LEN(MULTI$)=6 THEN 1850
1810 POSITION 14,14:? "Not possible";
1820 ? CHR$(253):FOR 0=1 TO 20:NEXT 0:POSITION 14,14:? "                ";:POSITION
24,6:? "            ";
1830 IF LEN(MULTI$)<6 THEN MULTI$(LEN(MULTI$)+1)=" ":GOTO 1830
1840 GOTO 1800
1850 DOB$=MULTI$
1860 POSITION 14,7:INPUT MULTI$:IF LEN(MULTI$)=0 THEN 1860
1870 IF LEN(MULTI$)<9 THEN 1910
1880 POSITION 12,14:? "Number too long";
1890 ? CHR$(253):FOR 0=1 TO 20:NEXT 0:POSITION 12,14:? "                    ";:POSITI
ON 14,7:? "                  ";
1900 GOTO 1860
1910 IF LEN(MULTI$)<8 THEN MULTI$(LEN(MULTI$)+1)=" ":GOTO 1910
1920 TELE$=MULTI$
1930 POSITION 6,8:INPUT MULTI$:IF LEN(MULTI$)=1 THEN 1950
1940 POSITION 6,8:? "           ";:GOTO 1930
1950 ZONE$=MULTI$
1960 POSITION 6,9:INPUT AD1$:IF LEN(AD1$)=0 THEN 1960
```

```
1970 POSITION 6,10:INPUT AD2$:IF LEN(AD2$)=0 THEN 1970
1980 POSITION 6,11:INPUT AD3$:IF LEN(AD3$)=0 THEN 1980
1990 AD1$(LEN(AD1$)+1)=AD2$:AD1$(LEN(AD1$)+1)=AD3$:IF LEN(AD1$)<61 THEN 2030
2000 POSITION 12,14:? "Address too long";:? CHR$(253):FOR Q=1 TO 20:NEXT Q
2010 POSITION 9,9:? "
                    ";
2020 GOTO 1960
2030 IF LEN(AD1$)<60 THEN AD1$(LEN(AD1$)+1)=" ":GOTO 2030
2040 ADD$=AD1$:RETURN
2050 WFILE$(J+1,J+16)=SURN$:WFILE$(J+17,J+18)=INIT$:WFILE$(J+19,J+24)=DOB$:WFILE
$(J+25,J+84)=ADD$
2060 WFILE$(J+85,J+92)=TELE$:WFILE$(J+93,J+93)=ZONE$:J=J+93:IF J=4650 THEN 2090
2070 GOSUB 180
2080 IF R=ASC(" ") THEN 1610
2090 GOSUB 310:GOTO 1410
2100 ? CHR$(125);CHR$(253):? :? :? :? :? :? :? "Please use a search routine":
? "to identify the file";
2110 FOR DEL=1 TO 250:NEXT DEL:GOTO 630
2120 REM FORMAT OUTPUT SUB ROUTINE
2130 TRAP 2100
2140 SURN$=WFILE$(I+1,I+15):INIT$=WFILE$(I+16,I+17):ADD$=WFILE$(I+24,I+83):DOB$=
WFILE$(I+18,I+23)
2150 TELE$=WFILE$(I+84,I+91):ZONE$=WFILE$(I+92,I+92)
2160 ? CHR$(125):POSITION 0,4:? "Surname: ";SURN$:POSITION 0,5:? "Initial: ";INI
T$
2170 POSITION 0,6:? "Date of birth(DDMMYY): ";:POUT$=DOB$(1,2):? POUT$;"/";:POUT
$=DOB$(3,4)
2180 ? POUT$;"/";:POUT$=DOB$(5,6):? POUT$
2190 POSITION 0,7:? "Phone number: ";TELE$;:POSITION 0,8:? "Zone: ";ZONE$;:POSIT
ION 0,9:? "Address: ";
2200 D=0:FOR P=1 TO 60:POUT$=ADD$(P,P):IF POUT$="," THEN 2220
2210 ? POUT$;:GOTO 2230
2220 D=D+1:? POUT$:POSITION 9,9+D
2230 NEXT P
2240 RETURN
2250 REM AGE SEARCH ROUTINE
2260 TRAP 630:PRINT CHR$(125)
2270 POSITION 15,1:? "Age search";:POSITION 7,11:? "Please enter todays date";:P
OSITION 10,13:? "(DDMMYY): ";:INPUT MULTI$
2280 FOR CTR=1 TO 6:IF MULTI$(CTR,CTR)<"0" OR MULTI$(CTR,CTR)>"9" THEN 2310
2290 NEXT CTR
2300 IF LEN(MULTI$)=6 THEN 2330
2310 POSITION 14,20:? "Not possible":? CHR$(253):FOR Q=1 TO 20:NEXT Q:POSITION 2
1,13:? "           ";
2320 GOTO 2270
2330 A$="":A$(LEN(A$)+1)=MULTI$(5,6):A$(LEN(A$)+1)=MULTI$(3,4):A$(LEN(A$)+1)=MUL
TI$(1,2)
2340 Z=VAL(A$):YOUNG=Z-60000:OLD=Z-120000:FOR I=1 TO J STEP 93:DOB$=WFILE$(I+18,
I+23)
2350 A$="":A$(LEN(A$)+1)=DOB$(5,6):A$(LEN(A$)+1)=DOB$(3,4):A$(LEN(A$)+1)=DOB$(1,
2):Y=VAL(A$)
2360 IF Y>YOUNG THEN 2390
2370 IF Y<OLD THEN 2420
2380 GOTO 2450
2390 GOSUB 2120:POSITION 15,14:? "Too young":GOSUB 2590:GOSUB 180
2400 IF R=ASC(" ") THEN 2450
2410 GOTO 630
2420 GOSUB 2120:POSITION 16,14:? "Too old":GOSUB 2590:GOSUB 180
2430 IF R=ASC(" ") THEN 2450
2440 GOTO 630
2450 NEXT I
2460 GOTO 630
2470 REM NAME SEARCH SUB ROUTINE
2480 PRINT CHR$(125)
2490 POSITION 14,1:? "Name search":POSITION 0,6:? "Please enter members surname
only.":POSITION 12,8:INPUT MULTI$
2500 VV=LEN(MULTI$)
2510 IF VV=0 THEN 2480
2520 IF VV<17 THEN 2550
```

```
2530 POSITION 13,14:? "Name too long";:? CHR$(253):FOR Q=1 TO 20:NEXT Q:POSITION
 13,14:? "                "
2540 POSITION 9,4:? "                                        ";:GOTO 2480
2550 FOR I=1 TO J STEP 93:IF MULTI$<>WFILE$(I+1,I+VV) THEN 2570
2560 GOSUB 2120:? :? :? :? :GOSUB 2590:GOSUB 180:IF R<>ASC(" ") THEN 630
2570 NEXT I
2580 ? CHR$(125):? CHR$(253):POSITION 6,14:? "No one of that name on file.":FOR
Q=1 TO 200:NEXT Q:GOTO 630
2590 ? "Do you want a hard copy. Y/N";:CLOSE #2:OPEN #2,4,0,"K:":GET #2,R:CLOSE
#2:IF R<>ASC("Y") THEN RETURN
2600 TRAP 2640
2610 OPEN #2,8,0,"P:":? #2;CHR$(27);"E"
2620 FOR Y=3 TO 14:? #2;CHR$(13);:FOR X=0 TO 39:LOCATE X,Y,R:? #2;CHR$(R);:NEXT
X:NEXT Y
2630 ? #2;CHR$(13):CLOSE #2:RETURN
2640 ? :? "         Printer not ready/":CLOSE #2:RETURN
2650 ? CHR$(125);CHR$(253):POSITION 0,10:? "No file present to save/":FOR DEL=1
TO 200:NEXT DEL:GOTO 630
2660 REM SAVE WFILE TO DISK
2670 PRINT CHR$(125):POSITION 15,1:? "SAVE FILE":POSITION 10,12:? "Insert save d
iskette";
2680 POSITION 7,18:? "Press 'E' to exit program";:GOSUB 180
2690 IF R<>ASC(" ") AND R<>ASC("E") THEN 630
2700 IF R=ASC("E") THEN END
2710 IF WFILE$="" THEN 2650
2720 ? CHR$(125):POSITION 5,8:? "ENTER FILE NAME"
2730 POSITION 6,10:? FILE$:POSITION 5,10:INPUT FILE$
2740 FILE$(11,14)=".DAT"
2750 OPEN #1,8,0,FILE$:? #1;J
2760 FOR S=1 TO J STEP 93:I$=WFILE$(S,S+92):? #1;I$:NEXT S:CLOSE #1:? CHR$(253):
GOTO 630
2770 REM LOAD FILE FROM DISK
2780 ? CHR$(125):OPEN #1,6,0,"D:*.DAT":? :? " AVAILABLE FILES":?
2790 TRAP 2810:INPUT #1,AD3$:IF AD3$(5,16)="FREE SECTORS" THEN 2810
2800 ? AD3$:GOTO 2790
2810 ? AD3$:CLOSE #1
2820 ? :? "DO YOU WISH TO LOAD(L) A FILE          OR START(S) A NEW FILE";:INPUT
A$
2830 IF A$="L" THEN 2860
2840 IF A$="S" THEN 1610
2850 GOTO 2820
2860 ? :? "ENTER FILE NAME":? " ";FILE$:? "";:INPUT FILE$
2870 FILE$(11,14)=".DAT"
2880 WFILE$="":OPEN #1,4,0,FILE$:INPUT #1;J:FOR S=1 TO J STEP 93:INPUT #1;I$:WFI
LE$(LEN(WFILE$)+1)=I$:NEXT S
2890 CLOSE #1:? CHR$(253):GOTO 630
```

SHARP MICRODEALER

SHARP MZ-3541

CPU	Z80A (two), 80C49
MEMORY	128K RAM, 8K ROM
LANGUAGE	Sharp BASIC
MASS STORAGE	Twin integral 5¼" floppy disk drives, total capacity 1.28 Mb
KEYBOARD	QWERTY, cursor, numeric pad, function keys
INTERFACES	RS-232C, Centronics, interface for extra external floppy disks
DISPLAY	Monochrome monitor, colour optional
GRAPHICS	80 by 25 text, 640 by 400 high-resolution graphics
SOUND	Single channel

Notes: The Sharp MZ-3541 is aimed at the businessman. RAM is expandable to 256K, while two disk drives may be added externally to complement the integral pair. Colour is only possible with the optional graphics expansion RAM. One Z80 handles the main CPU activities while the other handles peripheral activities. The third processor handles the keyboard. The availability of CP/M means a ready supply of business software.

LONDON

SHARPSOFT LTD.

Specialists in all Sharp software and hardware.

Sharpsoft Ltd, Crisallen House, 86-90 Paul Street, London EC2. Tel: 01 - 729 5588.

LANCASHIRE

STATION ELECTRICAL
Dept CT, Coastal Road, West Bank,
Lancs LH26HN. Tel: 0524 824519
Large range of software, books and peripherals for SHARP MZ 700. 2nd hand computers bought, sold and part exchange. Also repairs.
SAE for lists

MICRODEALER — ORDER FORM

Please include my business details in the next available issue of Computing Today:

Business Name: .....

Address: .....

.....

Details: .....

Tel. No.: .....

Open Hrs: .....

Contact (Office Use Only): .....

Post To: Computamart, Computing Today, 1 Golden Square, London W1.

ONLY £20.00

COMPUTING TODAY
CLASSIFIED ADVERTISEMENT — ORDER FORM

If you have something to sell now's your chance! Don't turn the page — turn to us!

Rates of charge: 40p per word per issue (minimum of 15 words). Please state classification and post to **COMPUTING TODAY, CLASSIFIED DEPT., 1 GOLDEN SQUARE, LONDON W.1.**

Please use BLOCK CAPITALS and include post codes.

Name (Mr/Mrs/Miss/Ms) .....
(delete accordingly)

Address .....

.....

.....

Signature ..... **Date** .....

Daytime Tel. No. .....

Please place my advert in **COMPUTING TODAY** for ..... issues commencing as soon as possible.

ACT MICRODEALER

xi APRICOT

CPU	8086
MEMORY	256K RAM
LANGUAGES	Microsoft BASIC, Personal BASIC
MASS STORAGE	No cassette drive Integral Sony 3½" 315K microfloppy disk drive Integral 5 or 10 Mb hard disk
OS	MS-DOS 2.11 with GSX bundled CP/M-86 (not yet available) Concurrent CP/M-86 (not yet available)
KEYBOARD	QWERTY, cursor, numeric pad, function keys
INTERFACES	RS-232C, Centronics, Microsoft mouse
DISPLAY	Monitor (supplied)
GRAPHICS	80 by 24 text with block graphics 800 by 400 high-res graphics under GSX
SOUND	No

Notes. The Apricot xi is a development of the award-winning Apricot, and replaces one of the latter's disk drives with an integral hard disk, providing vastly increased storage with faster access. Memory may be expanded in 128K increments to a maximum of 768K. The languages and operating systems mentioned above come bundled (except for Concurrent CP/M) and four software tools are also bundled, including an asynchronous package for use with the optional modem card.

HAMPSHIRE

TIMATIC SYSTEMS LTD
The Market, Fareham.
Tel: (0329) 239953

For the complete range of Apricot hardware and software. Also dealers for Zenith, Memotech. For future information call or ring anytime.

WALES

SIGMA SYSTEMS LTD
266 North Road, Cardiff
Tel: 0222 621414
Main dealer and Service for ACT, SIRIUS, APRICOT, IBM, COMMODORE & DIGITAL

MIDDLESEX

SIRIUS AND APRICOT IN STAINES
Micronomy Ltd., Unit 18, Central Trading Estate, Staines, Middlesex TW18 4XE.
TEL: STAINES 63651

WEST MIDLANDS

Q data limited
The Black Country's specialist in micro-computing. Full range of ACT Apricots and IBM personal computers.
The Limes, High Holborn, Sedgley, West Midlands.
Tel: Sedgley (09073) 62331

SCOTLAND

SIRIUS
is alive and well and supported at
ROBOX
(Office Equipment) Ltd,
The Scottish Computer Centre
Anderson Centre, Glasgow
041-221 8413/4
34 Queen Street, Edinburgh
031-225 3871

TO FILL THIS SPACE PHONE CAROLINE ON 01-437-0699

CBM MICRODEALER

Notes: The Commodore 64 is a popular micro with a great deal of games software available. There is also some business software available.

The Commodore 715B is the top model in the 700 range of business machines.

YORKSHIRE

YORKSHIRE ELECTRONICS
Commodore Appointed Commerical Systems Dealer
Full range of peripherals and Software available.
Caxton House, 17 Fountain St., Morley, West Yorkshire.
Tel: 0532 522181

NASCOM MICRODEALER

NASCOM 3

CPU	2 MHZ Z80	DISPLAY	40 or 80 column 25-line display
MEMORY	8K or 32K inbuilt RAM (expandable to 60K)		
LANGUAGE	Full Microsoft BASIC	GRAPHICS	High resolution graphics with 8 foreground and 8 background colours (400 x 256 pixels) Double density graphics with 2 colours (800 x 256 pixels)
MASS STORAGE	Single or twin 5.25" disc drives 350K capacity per drive		
OS	NAS-DOS or CP/M 2.2		
KEYBOARD	Full size QWERTY	SOUND	No
INTERFACES	RS232 and 16-bit parallel		

DEVON

PLYMOUTH'S DEALER nascom
S & R BREWSTER LIMITED
86-88 Union Street, Plymouth PL1 3HG
Tel: 0752 665011 Open: 6 days

IS YOUR COUNTY REPRESENTED? — IF NOT — WHY NOT?

LANCASHIRE

EV COMPUTING
700 Burnage Lane,
Manchester M19.
Tel: 061-431 4866
80-BUS SOLUTIONS

COMPUTING TODAY

Lineage: 40p per word.

Semi display: £9.00 per single column centimetre
Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01-437 0699

Send your requirements to:
CAROLINE
ASP LTD, 1 GOLDEN SQUARE,
LONDON W1

HARDWARE

NEWBRAIN & SANYO

Professional Micro Computers for the prices of hobby machines.

NEWBRAIN ON SPECIAL OFFER
£99 PLUS VAT
Model AD with free beginners guide and tape, investment software and UK postage. As above plus about **£100** value of business/general software **£129** + VAT.
Limited offer-ring now
SANYO 550/555 COMPUTERS
Micropro Wordstar, Calcstar etc at no extra cost! **Printers:** Epson, KDC, Juki, Kaga, Daisystep 2000 etc.
Monitors & Recorders.
Call STEVENAGE (0438) 812439
anytime for hardware/software lists
Mail Order and **Access** facilities
ANGELA ENTERPRISES
4 Ninnings Lane, Rabley Heath, Welwyn, Herts AL6 9TD.

AMSTRAD accessories, video and power extension lead:- £9.95 stereo amplifier, connect two speakers for stereo:- £15.95 inclusive. Amware, 15 Heath End Road, Flackwell Heath, Bucks, HP10 9DT.

SOFTWARE

VIC-20 USERS

A renumber facility for your basic programs requiring no memory expansion. Cassette £2.00 including post and package.
Dept 2, Bolling Software, PO Box 42, Bradford, West Yorkshire, BD4 7UD

COLOUR GENIE owners quiz game £2.99 coming soon — Quiz Master £4.99, Fires of Mordor, 41 Pexwood Rd, Tormorden, Lancs.

CONFUZION
BY INCENTIVE
£6.95

REPAIRS

MICROSERV for expert repairs to BBC machines and disk drive. For details call or write to Unit 4, Denny Workspace, Denny, Scotland. FK6 6DW. Tel: Denny (0324) 823468.

HOME COMPUTER REPAIRS

Look at our fantastic prices on repairs!

BBC B	£27.50
VIC 20	£20.00
COMMODORE 64	£27.50
DRAGON	£30.00
ORIC/ATMOS	£25.00
ZX SPECTRUM	£17.25
ZX INTERFACE	£17.25
ZX MICRODRIVE	£17.25

PLUS OTHERS!
The above prices are inclusive of parts, labour, P&P. All repairs carry 6 months warranty on replaced parts. Extended warranties, peripheral repairs, upgrades etc. etc. All available. Ring for full details (0234) 213645.
ZEDEM COMPUTER LTD
2 Kimbolton Rd, Bedford.

SOFTWARE

PROGRAMMER'S MARKET

This directory lists 700 software buyers in the USA looking for freelance microcomputer software. Each listing gives the contact; name, address, types of program worked, submission requirements, reoyalty/payment terms, sales tips, contract work available, and new information on technical writing opportunities. 343 pages. **Price £14.50**
Access/Visa accepted.
Henry Greenwood, 28 Great James Street, London WC1N 3HL.

COURSES

LEARN TO USE your computer on a weekend/holiday course. Details from:- Jaysoft MICRO Developments, 2 Wester Row, Greenlaw, Berwickshire.

SOFTWARE

** TAX MASTER! **

END YOUR INCOME TAX WORRIES!
A user friendly tape/disc, password protected, 2nd processor, etc. (view change and print-out operations) **£9.75**
ANDERSON'S
56 Waterloo Rd., Freemantle, Southampton.

TURBO PASCAL

Extended Pascal for PC DOS, MS DOS, CP/M£86 and CP/M-80, includes full screen editor, floating point arithmetic, full string handling feature, random access data files, complies faster than IBM or MT + Pascal, requires less than 35K of disk space, 250 page manual and FREE spreadsheet program written in Turbo Pascal.
"ONLY £54.95"
All prices fully inclusive for prepaid orders.
CONGUIN SOFTWARE,
14 GOODWOOD CLOSE, MORDEN, SURREY SM4 5AW.
No callers please Phone 0524 381423

T199/4A BASIC

Seven games on two tapes one price. Bagbounder Food of the Gods, 3D Maze Hunt, Eat-Up, Laundry Man, Lower and Higher and Rebound Sector **£6.95**. 50% PROFITS FOR ETHIOPIA. P&P free. Send cheques or PO's to:
KABEE SOFTWARE, 73 THORNES PARK, RASTRICK, BRIGHOUSE, W. YORKS HD6 3DA.

* UNLOCK YOUR AMSTRAD*

Amskey

EASY TO USE utility program. Removes protection and allows listing, studying and copying of your precious software.
YOU CHOOSE loading speed and protection of your back-up copies.
Only **£6.99 incl. p&p.** Overseas please add £1.00 postage.
Interlock Services Ltd. Dept C.T.
37B New Cavendish St, London W1M 8JR. Tel: 01-609 8301.

ALARMS

BURGLAR ALARM Equipment. Please visit our 2,000 sq. ft. showrooms or write or phone for your free catalogue. CWAS Ltd., 100 Rooley Avenue, Bradford BD6 1DB. Telephone: (0274) 731532.

FOR SALE

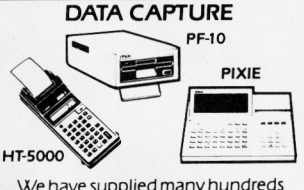
EPSON

As one of EPSONs largest dealers we sell *service* and *support* their full range. For quality, style and reliability they are hard to match. Write for our full list and ex-demo and specials list.

COMMUNICATIONS

Acoustic and Direct Connect Modems for all asynch speeds including electronic mail and viewdata. Prices from £79.

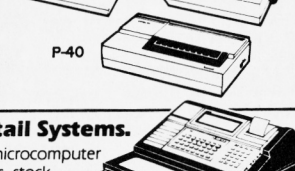
DATA CAPTURE



We have supplied many hundreds of terminals for data capture for sales men, shops or depots and can get your data into most micros, minis and mainframes over telephones. Call us to discuss your applications.

PRINTERS

We supply famous EPSON printers for all makes of computer. The LQ1500 for instance prints Near Letter Quality at 67cps and Draft at 200cps. The incredible P-40 costs only £87 (ex VAT) and prints up to 80 characters per line *and* BARCODES.



ACTIL Is a new concept in Retail Systems.

A general purpose, battery powered CP/M microcomputer which looks like a till, but also does accounts, stock control, order entry and communicates. Ideal for small shops, video libraries and market traders, as well as larger groups.

ACTEL
The Computer Centre,
Bugbrooke Road, GAYTON,
NORTHAMPTON NN7 3EU.
TEL: (0604) 858011.

VALUABLE ADVERTISING SPACE FOR SALE: RING CAROLINE ON 01 437 0699 EXT 323

COMPUTAMART

AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...

CHESHIRE

Computer Junk Shop

We Buy, Sell, Break Computers & Peripherals.
10 Waterloo Rd, Widnes, Halton. Tel: 051 420 4590.

IS YOUR COUNTY REPRESENTED? IF NOT, WHY NOT? TO ADVERTISE PHONE 01 437 0699

HERTFORDSHIRE

NEWBRAIN & SANYO
HARDWARE & SOFTWARE
Printers, Epson, KDC, Juki etc. Monitors, Tape Recorders, Books, Expansions, CP/M. **Sanyo 550/555 Computers.** Access/Mail Order. Ask for details.
ANGELA ENTERPRISES
Tel: Stevenage (0438) 812439 anytime

LONDON

LEABUS
legal and busines software
Specialists in wordprocessing systems (multi-lingual wordprocessors etc) based on the Apricot Computers.
Open 9am-6pm. Telephoe anytime.
114 Brandon Street, London SE17 1AL.
Telephone: 01 708 2756.

SOUTH LONDON

CROYDON COMPUTER CENTRE
Authorised Acorn Service Centre
29a Brigstock Rd., Thornton Heath, Surrey. Tel: 01 - 689 1280
BBC, Acorn, Electron, Genie, Oric, Kaga Microvitek Zenith Monitors. OKI 80, 82A + 84 Printers. Paper, Ribbons, Software etc. BUY-HIRE.

MIDDLESEX

SCREENS MICROCOMPUTERS
6 Main Ave., Moor Park, Northwood, Middx.
Tel: Northwood (09274) 20664
Telex: 923574 ALACOL G.
Official Dealers for: Acorn, Atari, Amstrad, Apricot, Commodore, Dragon, Einstein, Memotech, Oric, Psion, Sirius, Sanyo & Sinclair.
Open 6 days per week

NORFOLK

ANGLIA COMPUTER CENTRE
88 St Benedicts Street, Norwich.
Tel: (0603) 29652/26002.
Open: 6 days 9am-5.30pm.

SCOTLAND

VICTOR MORRIS GLASGOW
TANDY TRS 80, VIC 20, VIDEO GENIE, APPLE PANASONIC, CUMANA, EPSOM ETC.
340 Argyle Street, Glasgow G2: 041 221 8958

SUSSEX

GAMER
24 Gloucester Road, Brighton.
Tel: 0273-698424.
Open: Mon-Fri 10am-5.30pm, Sat 9am-5.30pm.

TYNE AND WEAR

HCCS ASSOCIATES
533 Durham Rd., Low Fell, Gateshead. Tel. Newcastle 821924.
Open: 6 days 9am-5.30pm (Sat 10am-5.30pm). Specialists in: Acorn, BBC, Video Genie, VIC 20.

CLASSIFIED ORDER FORM COMPUTAMART

Please include my business details in the next available issue of Computing Today:

Business Name: .....

Address: .....

.....

.....

Details: .....

.....

.....

.....

ONLY £17.50!

Tel. No.: .....

Open Hrs: .....

Contact (Office Use Only): .....

Post To: Computamart, Computing Today, 1 Golden Square, London W1.

Oxford Computer Systems (Software) Ltd.
Hensington Road, Woodstock, Oxford OX7 1JR, England
Telephone (0993) 812700 Telex 83147 Ref. OCSL

Compilers like these don't grow on trees

Oxford Pascal is Fast

Oxford Pascal compiles down to FAST COMPACT P-code, giving you the real speed and power of Pascal, together with the ability to compile very large programs.

Oxford Pascal is Standard

Oxford Pascal is a full extended implementation of Standard Pascal. This means that you can compile any Pascal program (subject to size), written on any computer, anywhere.

Oxford Pascal is Compact

Because it compiles into P-code, Oxford Pascal reduces programs into the most compact form possible. In fact it allows you to pack more code into your BEEB than any other language, and should your programs become too large, you can still use the CHAIN command to overlay limitless additional programs without losing data.

Graphics & Sound Extensions

In addition to the entire Pascal language, Oxford Pascal features a whole range of Graphics (all modes) and sound extensions designed to make maximum use of the BBC Computer. Oxford Pascal also provides numerous extensions such as hexadecimal arithmetic and bit manipulation instructions.

Oxford Pascal in Education

In Education, Oxford Pascal is fast becoming a *de facto* standard. It is already the most popular Pascal on the Commodore 64, and will soon be released for the Spectrum and the Amstrad. In fact, Oxford Pascal will soon be available for 90% of the computers installed in the U.K., and is already available in German, French, Swedish, and American versions. **Students and teachers alike find that it makes sense to use a standard implementation of Pascal across the whole range of educational micros.** Call us for details of our generous educational discounts.

Manual

Both these compilers come with a manual which has been carefully designed, not only as a quick reference guide, but also as a full tutorial for those new to Pascal.

Resident and Disc Compiler

Oxford Pascal comes in two forms:

For Tape Users...Oxford Resident Pascal. A compiler located largely in ROM which is available at any time. Programs can be written and compiled on the spot without disc or tape access, and compilation is fast enough to make using the compiler much like using the BASIC interpreter. Thus, learning Pascal is a simple interactive process. Some 15K of memory is available for user programs, the remainder being reserved for compiled object code.

For Disc Users...Oxford Disc Pascal offers all the above PLUS...a full disc compiler which is capable of using the WHOLE memory for Pascal object code, it is supplied with a powerful LINKER, allowing you to break large programming tasks down into separately compilable, easily-manageable files.

Friendly Error Messages

Many compilers produce little more than an error and line number to help correct mistakes in Pascal programs. Oxford Pascal however, gives you one of 49 friendly and informative error messages. Messages which not only indicate the reason for an error, but also print out the line in question with a pointer to the exact position where the error was detected.
Run-time errors are reported using line-numbers from the original source-program, with a full explanation of how the error occurred.

Powerful Editor

With Oxford Pascal there is no need for you to learn how to use a new Editor. Pascal programs can be entered in exactly the same way as BASIC programs, without the need to learn any new commands. When you are used to using Pascal, you will find our extensions to the Standard Editor even more useful. What is more, Oxford Pascal allows you to mix BASIC and Pascal together, in much the same way that you can mix BASIC and assembler. In fact you can, if required, mix all three together...BASIC, Pascal and assembler...in one program.

Stand Alone Code

Unlike other compilers, Oxford Disc Pascal allows you to compile on the BBC and then relocate your program so that it will run on any BBC. The relocated program will run without a Pascal **ROM** and can be loaded and run from tape or disc just like any other program.

This means that you can distribute or sell your software freely and without the need for **ROMs**, to run either of the above machines.

Price/availability matrix

	BBC 'B'	C64	SPECTRUM
DISC	£49.95	£49.95	Available July 1st 1985
CASSETTE	£39.95	£22.95	

All prices are inclusive of VAT
Please add £2.00 for postage and packing

Oxford Compilers — The Future

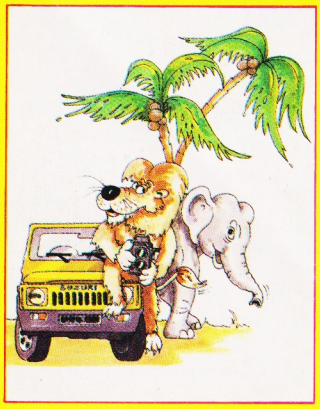
During the next year, we at Oxford will be releasing a series of language implementations such as C, and Modular 2, for the BBC, and other popular micros.

These compilers are being built, using the most modern techniques in **automated** compiler construction, and will bring to the micro-user, a level of robustness and efficiency, only now becoming available to mini and mainframe users.

Oxford...
the **Compiler Compilers.**

Oxford Pascal Order form. Please make cheques payable to OCSS Ltd
Please rush me my copy of Oxford Pascal. I enclose £ ..... including £2 postage and packing
My computer is BBC ☐ C64 ☐ (Please tick as appropriate)
I would like my compiler supplied on DISC ☐ CASSETTE ☐
Name .....
Address .....
Postcode .....
Telephone .....

WIN A SAFARI TO KENYA



An opportunity to win a super ELEPHANT Safari for two persons – luxury accommodation, travel and all expenses paid – with every purchase of a 10-disk ELEPHANT pack made between 1st April and 30th June 1985!
Look for your participating ELEPHANT Dealer's special Competition Display and pick up a simple puzzle card with your purchase. Send it to us with an ELEPHANT 10-pack top to receive, completely free, an ELEPHANT 'T' shirt and an opportunity in this exciting competition. Each subsequent 10-pack purchase and completed puzzle card gives you another opportunity – there's no limit!
Should you have difficulty in obtaining ELEPHANT products, phone or write for details of your nearest Dealer.

Dennison

ELEPHANT NEVER FORGETS

Dennison Manufacturing Co. Ltd.
Colonial Way, Watford, Herts WD2 4JY, Tel: Watford (0923) 41244, Telex: 923321

France: Soroclass, 45, rue de l'Est - 92100, Boulogne.
Tel. Réseau de Distribution: 605.98.99, Administration des Ventes: 605.70.78, Telex: EMS 206 436 E
Germany: Marcom Computerzubehör GmbH, Podbielskistr. 321, 3000 Hannover 51, Tel: (0511) 647420, Telex: 923818
Other Countries: Dennison International Company, 4006 Erkrath 1, Matthias-Claudius-Strasse 9, Telex: 858 6600